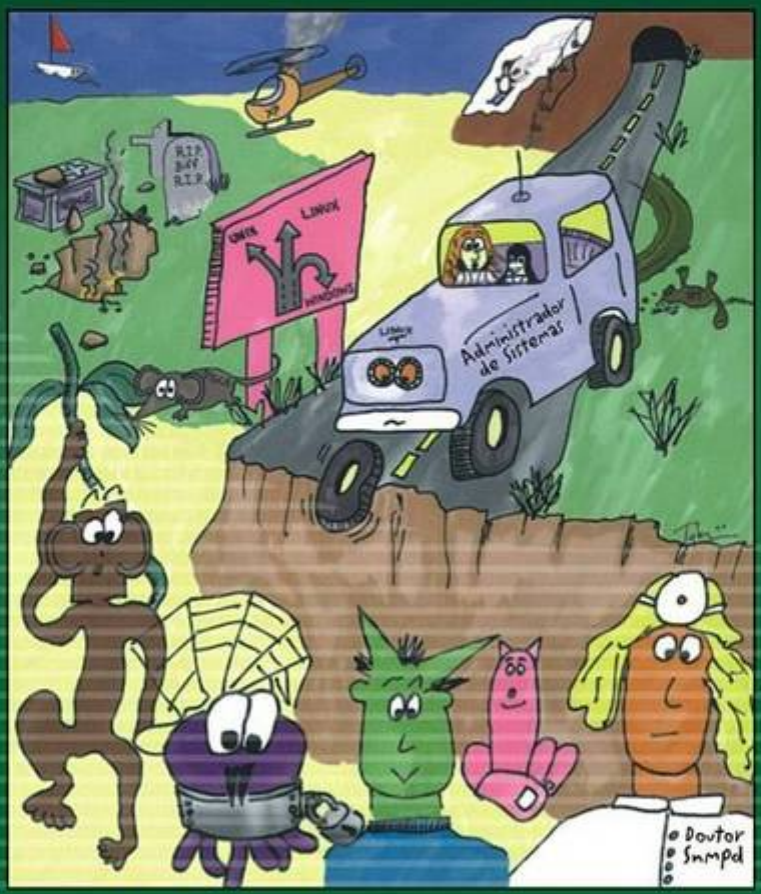
Snow666

MANUAL COMPLETO DO LINUX

Guia do administrador



PEARSON
Makron Books

EVI NEMETH / GARTH SNYDER / TRENT R. HEIN
Com Adam Boggs, Matt Crosby e Ned McClain

www.sn0w666.blogspot.co
Sn❄w666

Página em branco

www.snow666.blogspot.com
Snow666

Manual Completo do Linux

Evi Nemeth
Garth Snyder
Trent R. Hein
Com
Adam Boggs, Matt Crosby e Ned McClain

Tradução
Ariovaldo Griesi

Revisão técnica
Prof. Dr. Mário Olimpio de Menezes
Pesquisador do Instituto de Pesquisas Energéticas e Nucleares - IPEN/CNEN SP
Professor da FASP - Faculdades Associadas de São Paulo, UNIP, CEETESP e IPEN

São Paulo

Brasil Argentina Colômbia Costa Rica Chile Espanha Guatemala México Peru Porto Rico Venezuela

www.snow666.blogspot.com
Snow666

© 2004 by Pearson Education do Brasil
Título original: Linux Administration Handbook
© 2002 by Prentice Hall, uma empresa do grupo Pearson Education, Inc.
Tradução autorizada a partir da edição original em inglês, Linux Administration Handbook 1st Edition, by Nemeth, Evi; Snyder, Garth; Hein, Trent R., publicada pela Pearson Education, Inc, sob o selo Prentice Hall PTR

Todos os direitos reservados. Nenhuma parte desta publicação poderá ser reproduzida ou transmitida de qualquer modo ou por qualquer outro meio, eletrônico ou mecânico, incluindo fotocópia, gravação ou qualquer outro tipo de sistema de armazenamento e transmissão de informação, sem prévia autorização, por escrito, da Pearson Education do Brasil.

Diretor Editorial: José Martins Braga
Editora: Gisélia do Carmo Costa
Gerente de Produção: Heber Lisboa
Produtora Editoral: Marileide Gomes
Capa: Marcelo Françozo (*sobre o projeto original de Anthony Gemmellaro*)
Ilustração (desenhos a bico-de-pena): Tyler Curtain
Editoração Eletrônica: ERJ Composição Editorial e Artes Gráficas Ltda.
Impressão: São Paulo – SP

Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro, SP, Brasil)

Nemeth, Evi
Manual Completo do Linux / Evi Nemeth; Garth Snyder; Trent R. Hein ; com Adam Boggs, Matt Crosby e Ned McClain ; [tradução Ariovaldo Griesi ; revisão técnica Mario Olimpio de Menezes]. -- São Paulo : Pearson Makron Books, 2004

Título original: Linux Administration Handbook
Bibliografia
ISBN 85-346-1486-5

1. Linux (Sistema operacional de computador) 2. Redes de computadores I. Snyder, Garth. II . Hein, Trent R. III. Título.

04-1903 CDD-005.432

Índices para catálogo sistemático:

1. Linux : sistema operacional específico: Computadores : Processamento de dados 005.432

2005
1ª reimpressão
Direitos exclusivos para a língua portuguesa cedidos à
Pearson Education do Brasil,
uma empresa do grupo Pearson Education
Av. Ermano Marchetti, 1435
CEP: 05038-001 – Lapa – São Paulo – SP
Fone (11) 3613-1222 Fax (11) 3611-0444
e-mail: vendas@pearsoned.com

www.snow666.blogspot.com
Snow666

Sumário

Prefácio xxv

Introdução xxvii

Agradecimentos xxix

Sobre os autores xxx

Seção 1 – Administração Básica

Capítulo 1 Por Onde Começar 3

1.1 Background sugerido 3
1.2 O relacionamento do Linux com o UNIX 4
1.3 Histórico do Linux e do UNIX 4
1.4 Distribuições Linux 5
Portanto, qual é a melhor distribuição? 5
Ferramentas administrativas específicas a cada distribuição 6
1.5 Convenções tipográficas e de notação 6
Informações específicas de sistema 7
1.6 Onde buscar informações 7
Organização das páginas de manual 8
man:read manual pages 9
Outras fontes de informações sobre o Linux 9
1.7 Como localizar e instalar software 10
1.8 Tarefas essenciais de um administrador de sistemas 11
Adicionando e removendo usuários 12
Adicionando e removendo hardware 12
Realizando backups 12
Instalando software novo 12
Monitorando o sistema 12
Diagnosticando 12
Mantendo documentação local 12
Auditando a segurança 12
Ajudando os usuários 13
1.9 Administração de sistemas sob pressão 13
1.10 Leitura recomendada 14
1.11 Exercícios 14

Capítulo 2 Inicialização e Desligamento 15

2.1 Bootstrapping 15
Booting automático e manual 16
Etapas no processo de inicialização 16
Inicialização do kernel 16
Configuração de hardware 16
Processos de sistema 17
Intervenção do operador (somente para boot manual) 17
Execução de scripts de inicialização 17
Operação multiusuário 17

www.snow666.blogspot.com
Snow666

2.2 Inicializando PCs ... 18
Em que aspectos um PC é diferente de hardware proprietário ... 18
O processo de inicialização de um PC ... 18
2.3 Boot loaders: LILO e GRUB ... 19
LILO: o boot loader tradicional do Linux ... 19
GRUB: o loader GRand Unified Boot ... 20
Fazendo boot múltiplo em PCs ... 20
Pegadinhas de sistemas com boot múltiplo ... 21
Configuração de boot múltiplo com o LILO ... 21
2.4 Inicializando em modo monousuário ... 22
2.5 Scripts de inicialização ... 22
init e níveis de execução ... 23
Scripts de inicialização Red Hat ... 25
Scripts de inicialização SuSE ... 26
Scripts de inicialização Debian ... 28
2.6 Reinicializando e desligando ... 29
Delisgar a força elétrica ... 29
shutdown: a maneira gentil de desligar um sistema ... 30
halt: uma maneira mais simples de desligar um sistema ... 30
reboot: reinicialização rápida e suja ... 30
telinit: modifica o nível de execução de init ... 30
poweroff: solicita ao Linux para desligar o sistema ... 31
2.7 Exercícios ... 31

Capítulo 3 Poderes da Conta Root 32

3.1 Propriedade de arquivos e processos ... 32
3.2 O superusuário ... 33
3.3 Selecionando uma senha root ... 34
3.4 Tornando-se o root ... 34
su: identidade de usuário substituta ... 35
sudo: um su limitado ... 35
3.5 Outros pseudo-usuários ... 37
bin: proprietário legal dos comandos de sistema ... 37
daemon: proprietário de software de sistema não privilegiado ... 37
nobody: o usuário genérico NFS ... 37
3.6 Exercícios ... 37

Capítulo 4 Controlando Processos 38

4.1 Componentes de um processo ... 38
PID: número de identificação de processo ... 39
PPID: PID-pai ... 39
UID e EUID: identificadores de usuário real e efetivo ... 39
GID e EGID: identificadores de grupo real e efetivo ... 39
Gentileza ... 40
Terminal de Controle ... 40
4.2 O ciclo de vida de um processo ... 40
4.3 Sinais ... 40
4.4 Kill e killall: enviando sinais ... 42
4.5 Estados de processos ... 43
4.6 Nice e renice: influência sobre a prioridade de programação ... 43
4.7 ps: monitorar processos ... 44
4.8 top: monitoramento ainda melhor dos processos ... 46
4.9 Processos descontrolados ... 46
4.10 Exercícios ... 47

www.snow666.blogspot.com
Snow666

Capítulo 5 O Sistema de Arquivos 49

5.1 Nomes de caminhos ... 50
5.2 Montando e desmontando sistemas de arquivos ... 51
5.3 A organização da árvore de arquivos ... 52
5.4 Tipos de arquivos ... 53
Arquivos regulares ... 54
Diretórios ... 54
Arquivos de dispositivos de blocos e caracteres ... 55
Sockets de domínio local ... 56
Pipes com nomes ... 56
Links simbólicos ... 56
5.5 Atributos de arquivos ... 56
Os bits setuid e setgid ... 56
O bit sticky ... 57
Os bits de permissão ... 57
Visualizando atributos de arquivos ... 57
chmod: modificando permissões ... 59
chown: modificando propriedade e grupo ... 60
umask: atribuindo permissões-padrão ... 60
Flags adicionais ... 61
5.6 Exercícios ... 61

Capítulo 6 Adicionando Novos Usuários 63

6.1 O arquivo /etc/passwd ... 63
Nome de login ... 64
Senha criptografada ... 65
Número UID ... 66
Número GID Padrão ... 66
Campo GECOS ... 66
Diretório inicial ... 67
Shell de login ... 67
6.2 O arquivo /etc/shadow ... 67
6.3 O arquivo /etc/group ... 68
6.4 Adicionando usuários ... 69
Editando os arquivos passwd e shadow ... 70
Configurando uma senha inicial ... 70
Criando o diretório inicial do usuário ... 71
Copiando arquivos de inicialização padrão ... 71
Selecionando o diretório inicial de e-mail do usuário ... 72
Editando o arquivo /etc/group ... 72
Configurando cotas de disco ... 72
Verificando o novo login ... 72
Registrando a situação do usuário e informações para contato ... 73
6.5 Eliminando usuários ... 73
6.6 Desabilitando logins ... 73
6.7 Utilitários para gerenciamento de contas ... 74
6.8 Exercícios ... 75

Capítulo 7 Dispositivos Seriais 76

7.1 Padrões de dispositivos seriais ... 76
7.2 Conectores alternativos ... 79
A variante mini DIN-8 ... 79
A variante DB-9 ... 79
A variante RJ-45 ... 80

Snow666

O padrão Yost para cabeamento RJ-45 ... 81
7.3 Portadora via hardware e software ... 82
7.4 Controle de fluxo via hardware ... 83
7.5 Comprimento do cabo ... 83
7.6 Arquivos de dispositivos seriais ... 83
7.7 Setserial: informe ao driver sobre parâmetros de portas seriais ... 84
7.8 Configuração de software para dispositivos seriais ... 84
7.9 Configuração de terminais hardwired ... 85
O processo de login ... 85
O arquivo /etc/inittab ... 86
Suporte a terminais: os bancos de dados termcap e terminfo ... 87
7.10 Caracteres especiais e o driver de terminal ... 88
7.11 Stty: opções de configuração de terminais ... 88
7.12 Tset: configura opções automaticamente ... 89
7.13 Como desfazer a intrusão num terminal ... 89
7.14 Modems ... 90
Modulação, correção de erros e protocolos de compactação de dados ... 90
minicom: dial out ... 91
Modems bidirecionais ... 91
7.15 Depurando uma linha serial ... 91
7.16 Outras portas de E/S comuns ... 92
Portas paralelas ... 92
USB: o Barramento Serial Universal ... 93
7.17 Exercícios ... 94

Capítulo 8 Adicionando um Dfisco 95

8.1 Interfaces de disco ... 95
A interface SCSI ... 96
A interface IDE ... 99
O que é melhor, SCSI ou IDE? ... 100
8.2 Geometria do disco ... 101
8.3 Uma visão geral sobre o procedimento de instalação de discos ... 102
Conectando o disco ... 102
Formatando o disco ... 103
Identificando e particionando o disco ... 103
Estabelecendo volumes lógicos ... 104
Sistemas de arquivos do Linux ... 105
8.4 Os sistemas de arquivos ext2 e ext3 ... 106
Configurando extensões ext3fs ... 106
Configurando montagem automática ... 107
Habilitando o swapping ... 108
8.5 Fsck: verificação e reparo de sistemas de arquivos ... 109
8.6 Adicionando um disco ao Linux: um guia passo a passo ... 110
8.7 Exercícios ... 114

Capítulo 9 Processos Periódicos 116

9.1 Cron: agendando comandos ... 116
9.2 O formato de arquivos crontab ... 117
9.3 Gerenciamento de crontab ... 118
9.4 Empregos comuns para cron ... 119
Limpando o sistema de arquivos ... 119
Distribuição via rede de arquivos de configuração ... 120
Rodízio de arquivos log ... 120
9.5 Exercícios ... 120

www.snow666.blogspot.com
Snow666

Capítulo 10 Backups **121**

10.1 Práticas consagradas ... 121
Execute todos os dumps em uma máquina ... 122
Rotule suas fitas ... 122
Selecione um intervalo de backup razoável ... 122
Selecione sistemas de arquivos cuidadosamente ... 122
Faça backups diários caberem numa fita ... 122
Torne os sistemas de arquivos menores do que seu dispositivo de backup ... 123
Mantenha fitas fora da instalação ... 123
Proteja os seus backups ... 123
Limite a atividade durante os backups ... 123
Verifique suas fitas ... 124
Defina o ciclo de vida de uma fita ... 124
Projete seus dados para backups ... 124
Prepare-se para o pior ... 125
10.2 Dispositivos e mídia para backup ... 125
Disquetes ... 125
Superdisquetes ... 126
CD-R e CD-RW ... 126
Discos rígidos removíveis ... 126
Fitas-cartucho de 8 mm ... 126
Fitas DAT, cartuchos de 4mm ... 126
Fitas Travan ... 127
DLT ... 127
AIT ... 127
Mammoth ... 127
Jukeboxes, empilhadores e bibliotecas de fitas ... 127
Discos rígidos ... 128
Síntese dos tipos de mídia ... 128
O que comprar ... 129
10.3 Configurando um regime de backup incremental via dump ... 129
Fazendo o dump de sistemas de arquivos ... 129
Seqüências de dump ... 131
10.4 Restaurando dumps com restore ... 132
Restaurando arquivos individuais ... 132
Restaurando sistemas de arquivos inteiros ... 134
10.5 Realizando dump e restore para atualizações ... 134
10.6 Usando outros programas de arquivamento ... 135
tar: arquivos de pacotes ... 135
cpio: utilitário de arquivamento dos primórdios ... 135
dd: rodar bits ... 135
volcopy: sistemas de arquivos duplicados ... 136
10.7 Usando vários arquivos em numa única fita ... 136
Robô a seu dispor? ... 137
10.8 Amanda ... 137
A arquitetura do Amanda ... 137
Configuração do Amanda ... 138
O arquivo amanda.conf ... 138
O arquivo disklist ... 142
Arquivos de log do Amanda ... 143
Depuração do Amanda ... 143
Restauração de arquivos de um backup Amanda ... 146
Alternativas para o Amanda: outros pacotes de backup com fonte aberto ... 147
10.9 Produtos comerciais para backup ... 147
ADSM/TSM ... 148

www.snow666.blogspot.com
Snow666

Veritas ... 148
Legato ... 148
Outras alternativas ... 149
10.10 Leitura recomendada ... 149
10.11 Exercícios ... 149

Capítulo 11 Syslog e Arquivos de Log **151**

11.1 Políticas de registro ... 151
Jogando fora arquivos de log ... 151
Rodízio de arquivos log ... 152
Arquivando arquivos de log ... 153
11.2 Arquivos de log no Linux ... 153
Arquivos de log especiais ... 154
Registro do horário de inicialização e do kernel ... 154
11.3 Logrotate: gerenciando arquivos de log ... 155
11.4 Syslog: o registrador de eventos do sistema ... 156
Configurando syslogd ... 157
Exemplos de arquivos de configuração ... 159
Exemplo de saída de syslog ... 161
Desenvolvendo um esquema de registro em sua instalação ... 162
Software que usa syslog ... 162
Depurando o syslog ... 163
Utilizando syslog por intermédio de programas ... 163
11.5 Condensando arquivos de log em informações úteis ... 164
11.6 Exercícios ... 165

Capítulo 12. Drivers e o kernel **166**

12.1 Adaptação do kernel ... 167
12.2 Por que configurar o kernel? ... 167
12.3 Métodos de configuração ... 167
12.4 Ajustando um kernel Linux ... 167
12.5 Adicionando drivers de dispositivos ... 168
Números de dispositivos ... 169
12.6 Adicionando um driver de dispositivo Linux ... 170
12.7 Arquivos de dispositivos ... 171
Convenções de nomes para dispositivos ... 172
12.8 Módulos de kernel carregáveis ... 172
12.9 Construindo um kernel Linux ... 174
Construindo o binário do kernel Linux ... 175
12.10 Não conserte se não está quebrado ... 175
12.11 Leitura recomendada ... 176
12.12 Exercícios ... 176

Seção 2 – Redes de Computadores

Capítulo 13 Redes TCP/IP **179**

13.1 O TCP/IP e a internet ... 180
Um breve histórico ... 180
Como a Internet é gerenciada atualmente ... 180
Documentação e padrões de rede ... 181
13.2 Mapa rodoviário sobre redes ... 182
13.3 Pacotes e encapsulamento ... 183
A camada de link ... 184
Endereçamento de pacotes ... 185

Snow666

Portas ... 186
Tipos de endereço ... 186
13.4 Endereços IP: os detalhes viscerais ... 187
Classes de endereços IP ... 187
Sub-redes e máscaras de rede ... 187
A crise de endereços IP ... 189
CIDR ... 190
Alocação de Endereços ... 192
Endereços privados e o NAT ... 192
Endereçamento IPv6 ... 194
13.5 Roteamento ... 196
Tabelas de roteamento ... 196
Redirecionamentos ICMP ... 197
13.6 ARP: o protocolo de determinação de endereços ... 197
13.7 Adicionando uma máquina a uma rede ... 198
Atribuindo nomes de host e endereços IP ... 199
ifconfig: configura interfaces de rede ... 200
mii-tool: configura autonegociação e outras opções específicas da mídia ... 202
route: configuração de rotas estáticas ... 203
Rotas-padrão ... 204
Configurando DNS ... 205
A pilha de rede do Linux ... 206
13.8 Configuração de redes específicas de cada distribuição ... 206
Configuração de Rede para o Red Hat ... 206
Configuração de rede para o SuSE ... 207
Configuração de rede para o Debian ... 208
Configuração de rede com uma Interface Gráfica com o Usuário ... 209
13.9 DHCP ... 209
Software DHCP ... 210
Funcionamento do DHCP ... 210
Servidor DHCP do ISC ... 211
Configuração DHCP no Red Hat ... 212
Configuração DHCP no SuSE ... 212
Configuração DHCP no Debian ... 213
13.10 Reconfiguração e ajuste dinâmico do Linux ... 213
13.11 Questões de segurança ... 214
Encaminhamento IP ... 215
Redirecionamentos ICMP ... 215
Roteamento de origem ... 215
Pings transmitidos amplamente e outras formas de difusão ampla direcionada ... 215
Falsificação de IP (IP spoofing) ... 215
Firewalls baseados em hosts ... 216
Redes Privadas Virtuais - VPNs ... 216
Variáveis de kernel relacionadas com segurança ... 216
13.12 NAT no Linux (mascaramento de IP) ... 217
13.13 PPP: o protocolo ponto-a-ponto ... 218
Resolvendo problemas de desempenho no PPP ... 218
Conectando-se a uma rede via PPP ... 219
Fazendo um host falar PPP ... 219
Controlando links PPP ... 219
Localizando um host para dialogar ... 219
Atribuindo um endereço ... 219
Roteamento ... 220
Garantindo a segurança ... 220
Usando servidores de terminais ... 220

Snow666

Usando "chat scripts" ... 220
Configuração PPP no Linux ... 220
13.14 Peculiaridades de rede do Linux ... 226
13.15 Leitura recomendada ... 227
13.16 Exercícios ... 228

Capítulo 14 Roteamento **229**

14.1 Encaminhamento de pacotes: uma visão mais detalhada ... 229
14.2 Daemons de roteamento e protocolos de roteamento ... 231
Protocolos de vetor de distância ... 232
Protocolos de estado de link ... 232
Métrica de custo ... 233
Protocolos internos e externos ... 233
14.3 Protocolos na parada ... 234
RIP (Routing Information Protocol) ... 234
RIP-2: Routing Information Protocol, versão 2 ... 234
OSPF: Open Shortest Path First ... 234
IGRP e EIGRP: Interior Gateway Routing Protocol ... 235
IS-IS: o "padrão" ISO ... 235
MOSPF, DVMRP e PIM: protocolos de roteamento multidifusão ... 235
RDP ... 235
14.4 Routed: abra novos caminhos por conta própria ... 235
14.5 Gated: um daemon de roteamento melhor ... 236
Início e controle do gated ... 236
Tracing ... 237
O arquivo de configuração de gated ... 237
Instruções de configuração de opções ... 238
Definições de interfaces de rede ... 238
Outras definições variadas ... 240
Configuração de protocolo RIP ... 241
Algumas informações preliminares sobre OSPF ... 241
Configuração de protocolo para o OSPF ... 242
Configuração de protocolo para redirecionamentos ICMP ... 243
Rotas estáticas ... 244
Rotas exportadas ... 244
Um exemplo de configuração gated completo ... 245
14.6 Critérios de seleção de estratégia de roteamento ... 246
14.7 Roteadores Cisco ... 247
14.8 Leitura recomendada ... 249
14.9 Exercícios ... 250

Capítulo 15 Hardware de Rede **251**

15.1 LAN, WAN ou MAN? ... 251
15.2 Ethernet: a LAN comum ... 252
Funcionamento da Ethernet ... 252
Topologia Ethernet ... 253
UTP ... 253
Conectando e Expandindo redes Ethernet ... 255
15.3 Wireless: a LAN do nômade ... 257
15.4 FDDI: a LAN decepcionante e cara ... 258
15.5 ATM: a LAN prometida (porém altamente malograda) ... 259
15.6 Frame relay: a WAN sacrificatória ... 260
15.7 ISDN: a WAN nativa ... 260
15.8 DSL e modems a cabo: a WAN das pessoas ... 261

Snow666

15.9 Para onde as redes estão caminhando? ... 261
15.10 Teste e depuração de redes ... 262
15.11 Fiação predial ... 262
Opções de cabeamento UTP ... 262
Conexões para escritórios ... 262
Padrões de fiação ... 263
15.12 Questões de projeto ... 263
Arquitetura de rede *versus* arquitetura do prédio ... 264
Redes existentes ... 264
Expansão ... 264
Congestionamento ... 264
Manutenção e documentação ... 265
15.13 Questões de gerenciamento ... 265
15.14 Fornecedores recomendados ... 265
15.15 Leitura recomendada ... 266
15.16 Exercícios ... 267

Capítulo 16 O Sistema de Nomes de Domínios 268

16.1 DNS para o impaciente: acrescentando uma nova máquina ... 268
16.2 A história do DNS ... 269
16.3 Quem precisa do DNS? ... 270
16.4 Novidades no DNS ... 271
16.5 O espaço de nomes DNS ... 272
Mestres de seus domínios ... 274
Selecionando um nome de domínio ... 274
O inchamento dos domínios ... 275
Registrando um nome de domínio de segundo nível ... 275
Criando seus próprios subdomínios ... 275
16.6 O software BIND ... 276
Versões do BIND ... 276
Descobrindo que versão você possui ... 276
Componentes de BIND ... 277
named: o servidor de nomes BIND ... 277
Servidores autorizados e somente para caching ... 278
Servidores recursivos e não recursivos ... 279
A biblioteca solucionadora ... 279
Interfaces de shell para o DNS ... 280
16.7 Funcionamento do DNS ... 280
Delegação ... 280
Caching e eficiência ... 281
O protocolo DNS estendido ... 281
16.8 Questões Relacionadas a Clientes no BIND ... 282
Configuração do solucionador ... 283
Testes de solucionadores ... 284
Impacto sobre o restante do sistema ... 284
16.9 Configuração de servidores BIND ... 285
Exigências de Hardware ... 285
Inicialização de named ... 285
Arquivos de configuração ... 286
A instrução include ... 287
A instrução options ... 287
A instrução acl ... 291
A instrução server ... 292
A instrução logging ... 292
A instrução zone ... 292

www.snow666.blogspot.com
Snow666

A instrução key ... 295
A instrução trusted-keys ... 295
A instrução controls ... 295
DNS subdividido e a instrução view do BIND 9 ... 297
16.10 Exemplos de configuração BIND ... 298
Um computador doméstico com Linux ... 298
Uma pequena empresa especializada em segurança de sistemas ... 300
Um departamento de uma universidade ... 302
16.11 O banco de dados DNS ... 306
Registros de recursos ... 306
O registro SOA ... 308
Registros NS ... 309
Registros A ... 310
Registros PTR ... 310
Registros MX ... 311
Registros CNAME ... 313
O truque CNAME ... 313
Registros LOC ... 314
Registros SRV ... 315
Registros TXT ... 316
Registros de recursos no IPv6 ... 316
Registros de endereços IPv6 ... 317
Registros reversos no IPv6 ... 317
Comandos em arquivos de zonas ... 319
A zona de localhost ... 321
Registros de adesão: links entre zonas ... 321
16.12 Atualizando arquivos de zonas ... 323
Transferências de zonas ... 323
Atualizações dinâmicas ... 324
16.13 Questões de segurança ... 325
Retornando às listas de controle de acesso ... 326
Confinando named ... 327
Comunicação servidor-servidor segura com TSIG e TKEY ... 328
DNSSEC ... 329
Microsoft ruim, Linux bom ... 333
16.14 Testando e depurando ... 334
Registro ... 334
Níveis de depuração ... 338
Depurando com o ndc ... 338
Estatísticas no BIND 8 ... 339
Estatísticas no BIND 9 ... 340
Depurando com nslookup, dig e host ... 341
Delegações defeituosas ... 343
16.15 Finalizações dispersas ... 344
O arquivo de dicas ... 344
Configuração de localhost ... 345
Ferramentas para gerenciamento de hosts ... 345
DNS para sistemas que não estão na Internet ... 345
16.16 Informações específicas sobre distribuições ... 345
Versões ... 345
Arquivos BIND ... 346
O arquivo de comutação de servidor de nomes ... 346
Arquivos de configuração ... 346
16.17 Leitura recomendada ... 347
Mailing lists e grupos de notícias ... 347

Snow666

Livros e documentação diversa ... 347
Recursos on-line ... 347
As RFCs ... 348
16.18 Exercícios ... 349

Capítulo 17 O Sistema de Arquivos de Rede 351

17.1 Informações genéricas sobre o NFS ... 351
Versões do protocolo NFS ... 351
Escolha de transporte ... 352
Bloqueio de arquivos ... 352
Cotas de disco ... 352
Cookies e montagem sem estado ... 352
Convenções para atribuição de nomes para sistemas de arquivos ... 353
Segurança e NFS ... 353
Acesso root e a conta nobody ... 353
17.2 NFS no lado servidor ... 354
O arquivo exports ... 355
nfsd: servir arquivos ... 356
17.3 NFS no lado cliente ... 356
Montando sistemas de arquivos remotos no momento da inicialização ... 358
Restrições de portas seguras ... 358
17.4 nfsstat: fazendo o dump de estatísticas NFS ... 358
17.5 Servidores de arquivos NFS dedicados ... 359
17.6 Montagem automática ... 359
17.7 automount ... 360
O arquivo mestre ... 361
Arquivos de mapas ... 361
Mapas executáveis ... 361
17.8 amd: um automontador mais sofisticado ... 362
Mapas amd ... 362
Iniciando amd ... 363
Interrompendo amd ... 364
17.9 Leitura recomendada ... 364
17.10 Exercícios ... 364

Capítulo 18 Compartilhando Arquivos de Sistema 365

18.1 O que deve ser compartilhado ... 366
nscd: coloque em cache os resultados das pesquisas ... 366
18.2 Copiando arquivos ... 367
rdist: distribuição de arquivos ... 367
rsync: transferência de arquivos de maneira mais segura ... 369
Puxando arquivos ... 371
18.3 NIS: network information service ... 371
Grupos de rede ... 372
Priorizando fontes de informações administrativas ... 373
Vantagens e desvantagens do NIS ... 374
Funcionamento do NIS ... 374
Configurando um domínio NIS ... 376
18.4 NIS+: filho de NIS ... 377
18.5 LDAP: lightweight directory access protocol ... 378
Documentação e especificações do LDAP ... 379
Mãos na massa com o LDAP ... 379
Configurando um servidor OpenLDAP ... 380
Preenchendo o seu servidor ... 380

www.snow666.blogspot.com
Snow666

Configurando um cliente LDAP ... 381
LDAP e segurança ... 381
18.6 Exercícios ... 381

Capítulo 19 E-Mail **382**

19.1 Sistemas de e-mail ... 383
Agentes de usuário ... 384
Agentes de transporte ... 385
Agentes de entrega ... 386
Armazenamento de mensagens ... 386
Agentes de acesso ... 386
Agentes de submissão de e-mails ... 386
19.2 A anatomia de uma mensagem de e-mail ... 387
Endereços de e-mails ... 387
Lendo os cabeçalhos de mensagens de e-mail ... 388
19.3 Filosofia do e-mail ... 391
Usando servidores de e-mail ... 391
Utilizando centrais de e-mail (mail homes) ... 393
Utilizando IMAP ou POP ... 393
19.4 Nomes alternativos de e-mail ... 394
Obtendo listas de discussão de arquivos ... 396
Enviando e-mails a arquivos ... 396
Enviando e-mails para programas ... 397
Exemplos de nomes alternativos ... 397
Encaminhamento de e-mails ... 398
O banco de dados de nomes alternativos hashed ... 399
Listas de discussão e software de discussão ... 400
LDAP: lightweight directory access protocol ... 403
19.5 sendmail: o diretor de espetáculos do circo de e-mail ... 404
A história de sendmail ... 405
Versões fornecidas pelos fabricantes de sendmail ... 405
Instalação de sendmail em sendmail.org ... 406
Instalando o sendmail em sistemas Debian ... 408
O arquivo de comutação ... 408
Modos de operação ... 408
A fila de e-mails ... 409
19.6 Configuração de sendmail ... 411
Utilizando o pré-processador m4 ... 412
As peças de configuração de sendmail ... 413
Criando um arquivo de configuração com base em um modelo de arquivo .mc ... 413
Alterando a configuração de sendmail ... 414
19.7 Primitivas de configuração básicas para sendmail ... 414
A macro VERSIONID ... 414
A macro OSTYPE ... 415
A macro DOMAIN ... 416
A macro MAILER ... 417
19.8 Primitivas de configuração para sendmail mais incrementadas ... 417
A macro FEATURE ... 417
O recurso use_cw_file ... 418
O recurso redirect ... 418
O recurso always_add_domain ... 418
O recurso nocanonify ... 419
Tabelas e bancos de dados ... 419
O recurso mailertable ... 420
O recurso genericstable ... 421

Snow666

O recurso virtusertable ... 421
O recurso ldap_routing ... 422
Mascaramento e a macro MASQUERADE_AS ... 423
As macros MAIL_HUB e SMART_HOST ... 424
Mascaramento e roteamento ... 424
O recurso nullclient ... 425
Os recursos local_lmtp e smrsh ... 426
O recurso local_procmail ... 426
As macros LOCAL_* ... 426
Opções de configuração ... 426
19.9 Exemplos de arquivos de configuração ... 428
Um computador doméstico de um estudante de computação ... 428
Uma empresa pequena mas muito bem informada em termos de sendmail ... 430
Um outro exemplo mestre/cliente ... 433
A configuração de sendmail no Red Hat ... 435
Configuração de sendmail no SuSE ... 436
A configuração de sendmail no Debian ... 439
19.10 Recursos relacionados com spam no sendmail ... 441
Retransmissão ... 442
O banco de dados de acesso ... 444
Colocando usuários ou sites na lista negra ... 446
Verificação de cabeçalhos ... 447
Miltering: filtragem de e-mails ... 448
Manipulando spam ... 448
Exemplos de spam ... 449
SpamAssassin ... 451
19.11 Segurança e o sendmail ... 452
Propriedades ... 452
Permissões ... 453
E-mails mais seguros para arquivos e programas ... 453
Opções relacionadas com privacidade ... 454
Executando um sendmail onde se aplica chroot (para o verdadeiramente paranóico) ... 455
Ataques de recusa a serviços ... 455
Falsificações ... 456
Privacidade de mensagens ... 457
SASL: Simple Authentication and Security Layer ... 457
19.12 O desempenho do sendmail ... 458
Modos de entrega ... 458
Grupos de filas e divisão de envelopes ... 458
Executores de filas ... 459
Controles de carga média ... 459
Mensagens na fila que não podem ser entregues ... 459
Ajustando o kernel ... 460
19.13 Estatísticas, testes e depuração no sendmail ... 461
Testando e depurando ... 461
Entrega prolixa ... 462
Falando em SMTP ... 463
Registro ... 463
19.14 O sistema de e-mail Exim ... 465
Histórico ... 465
Exim no Debian ... 465
Configuração do Exim ... 465
Similaridades entre o Exim e o sendmail ... 466
19.15 Leitura recomendada ... 466
19.16 Exercícios ... 467

www.snow666.blogspot.com
Snow666

Capítulo 20 Gerenciamento e Depuração de Redes **469**

20.1 Diagnosticando uma rede ... 470
20.2 Ping: verifique se um host está funcionando ... 471
20.3 Traceroute: rastreia pacotes IP ... 472
20.4 Netstat: obtenha toneladas de estatísticas de rede ... 473
Monitorar o estado das conexões de rede ... 474
Inspecionando as informações de configuração de interface ... 474
Examinando a tabela de roteamento ... 475
Visualizando estatísticas operacionais para vários protocolos de rede ... 476
20.5 Analisadores de pacotes ... 476
tcpdump: o rei dos analisadores ... 477
Ethereal: analisador visual ... 477
20.6 Protocolos de gerenciamento de redes ... 478
20.7 SNMP: simple network management protocol ... 479
Organização SNMP ... 479
Operações de protocolos SNMP ... 480
RMON: MIB para monitoramento remoto ... 480
20.8 O agente NET-SMNP ... 481
20.9 Aplicativos para gerenciamento de redes ... 481
As ferramentas NET-SNMP ... 482
MRTG: Multi-Router Traffic Grapher ... 483
NOCOL: Network Operation Center On-Line ... 483
Plataformas comerciais para gerenciamento ... 484
20.10 Leitura recomendada ... 484
20.11 Exercícios ... 485

Capítulo 21 Segurança **486**

21.1 O Linux é seguro? ... 486
21.2 Segurança no Linux, a versão cliffsnotes ... 487
Filtragem de pacotes ... 487
Serviços desnecessários ... 487
Correções de software ... 487
Backups ... 488
Senhas ... 488
Vigilância ... 488
Filosofia geral ... 488
21.3 Como a segurança é comprometida ... 489
21.4 Problemas de segurança nos arquivos /etc/passwd e /etc/shadow ... 489
Verificação e seleção de senhas ... 489
PAM: spray culinário ou prodígio de autenticação? ... 490
Senhas-sombra (shadow) ... 491
Logins de grupos e logins compartilhados ... 491
Expiração da senha ... 491
Shells de usuários ... 492
Entradas root ... 492
21.5 Programas setuid ... 492
21.6 Permissões de arquivos importantes ... 493
21.7 Problemas de segurança variados ... 493
Registro de eventos remoto ... 493
Terminais seguros ... 493
/etc/hosts.equiv and ~/.rhosts ... 494
rexecd e tftpd ... 494
fingerd ... 494
Segurança e NIS ... 494

www.snow666.blogspot.com
Snow666

Segurança e NFS ... 495
Segurança e sendmail ... 495
Segurança e backups ... 495
Cavalos de Tróia ... 495
21.8 Ferramentas de segurança poderosas ... 495
nmap: varre portas de rede ... 496
ndiff: crie linhas de referência baseadas em nmap e procure encontrar alterações suspeitas ... 497
SAINT: verifica a eventual existência de vulnerabilidades de sistemas ligados em rede ... 497
Nessus: o varredor de redes de próxima geração ... 498
crack: localize senhas inseguras ... 498
tcpd: proteja serviços Internet ... 498
COPS: audit system security ... 499
tripwire: monitora alterações em arquivos de sistema ... 499
Ferramentas forenses ... 500
21.9 Ferramentas de segurança criptográficas ... 501
Kerberos: uma abordagem unificada à segurança de redes ... 501
PGP: Pretty Good Privacy ... 502
SSH: Secure SHell ... 502
OPIE: One-time Passwords In Everything ... 503
Identificadores de hardware ... 504
21.10 Firewalls ... 504
Firewalls para filtragem de pacotes ... 504
Como os serviços são filtrados ... 505
Firewalls proxy de serviços ... 505
Firewalls de inspeção com estados ... 506
Firewalls: qual o verdadeiro nível de segurança oferecido por eles? ... 506
21.11 Recursos de firewall no Linux: tabelas IP ... 506
21.12 VPNS (virtual private networks) ... 509
Túneis IPSEC ... 510
Tudo o que eu preciso é uma VPN, não é mesmo? ... 510
21.13 Fontes de informações sobre segurança ... 510
CERT: uma marca de serviços registrada da Carnegie Mellon University ... 511
SecurityFocus.com e a lista de discussão BugTraq ... 511
Boletim Crypto-Gram ... 511
SANS: System Administration, Networking, and Security Institute ... 511
Recursos de segurança específicos a cada distribuição ... 511
Outras listas de discussão e sites Web ... 512
21.14 Distribuições Linux fortificadas ... 512
21.15 O que fazer quando sua instalação tiver sido atacada ... 512
21.16 Leitura recomendada ... 514
21.17 Exercícios ... 514

Capítulo 22 Hospedagem Web e Servidores Internet 516

22.1 Hospedagem Web ... 516
22.2 Fundamentos sobre hospedagem Web ... 517
Localizadores de recurso uniforme ... 517
Como o HTTP funciona ... 518
Scripting CGI: gerando conteúdo dinamicamente ... 518
Balanceamento de carga ... 519
22.3 Instalação de servidores HTTP ... 519
Escolhendo um servidor ... 519
Instalando o Apache ... 520
Configurando o Apache ... 521
Executando o Apache ... 521

www.snow666.blogspot.com
Snow666

Hospedagem de alto desempenho ... 521
22.4 Interfaces virtuais ... 521
Configurando interfaces virtuais ... 522
Informando o Apache sobre a existência de uma interface virtual ... 523
22.5 Caching e servidores proxy ... 523
Configurando o Squid ... 524
22.6 Configuração de servidor FTP anônimo ... 524
22.7 Exercícios ... 526

Seção 3 – Um Bocado de Coisas

Capítulo 23 Instalação e Localização de Software 529

23.1 Instalação básica do Linux ... 529
23.2 Instalação automática ... 530
Inicialização de PCs via rede (netbooting) ... 530
PXE: o padrão para inicialização de PCs via rede ... 530
Configurando PXE para Linux ... 531
Fazendo a inicialização via rede de máquinas diferentes de PCs ... 531
Kickstart: Instalador automático do Red Hat ... 531
YaST: a ferramenta de instalação do SuSE ... 533
SystemImager ... 535
23.3 Localização ... 537
Automação ... 537
Criando versões de sua configuração ... 537
Usuários e departamentos rebeldes ... 538
Testes ... 539
Fazendo alterações irreversíveis ... 539
Controle de revisão ... 540
RCS ... 540
Documentação ... 542
Onde instalar software local ... 542
Quanto instalar nas máquinas-cliente ... 543
23.4 Mantendo seus sistemas atualizados com rsync ou rdist ... 543
23.5 Administração de pacotes ... 545
Pacotes RPM ... 546
Pacotes Debian ... 546
Automatizando a instalação de pacotes ... 547
Red Hat Network ... 547
23.6 Apt-get: download e instalação automáticos ... 548
Configurando apt-get ... 549
Um exemplo de arquivo /etc/apt/sources.list ... 549
Usando proxies para ter escalabilidade em apt-get ... 550
Configurando um servidor APT interno ... 550
Automatizando apt-get ... 551
23.7 Leitura recomendada ... 551
23.8 Exercícios ... 551

Capítulo 24 Impressão 553

24.1 Miniglossário de termos de impressão ... 553
24.2 Sistemas de impressão Linux ... 554
24.3 Tipos de impressoras ... 555
Impressoras seriais e paralelas ... 555
Impressoras de rede ... 555
A vida sem o PostScript ... 556

www.snow666.blogspot.com
Snow666

24.4 LPD: o velho e bom sistema de impressão ... 556
Uma visão geral do processo de impressão ... 556
Controlando o ambiente de impressão ... 557
lpd: o spooler de impressão ... 557
lpr: submeter tarefas de impressão ... 558
lpq: visualizar a fila de impressão ... 558
lprm: eliminar tarefas de impressão ... 558
lpc: fazer alterações administrativas ... 559
O arquivo /etc/printcap ... 560
Variáveis printcap ... 561
Variáveis printcap para dispositivos seriais ... 563
Extensões de printcap ... 564
Imprimindo em algo que não seja uma impressora ... 565
24.5 LPRNG ... 565
Os comandos LPRng ... 565
/etc/lpd.conf: configurar lpd ... 566
/etc/lpd.perms: configurar controle de acesso ... 566
Configurando o arquivo printcap ... 567
Filtros ... 567
Contabilidade ... 567
24.6 Adicionando uma impressora ... 567
Detalhes específicos a cada distribuição ... 569
24.7 Depurando problemas de impressão ... 569
24.8 Software comum para impressão ... 570
ghostscript ... 570
mpage ... 570
enscript ... 570
24.9 Filosofia de impressão ... 570
Contabilidade do uso de impressoras ... 570
Use páginas-banner somente quando necessário ... 570
Forneça lixeiras ... 571
Forneça pré-visualizadores ... 571
Compre impressoras baratas ... 571
Assegure sua impressora ... 571
24.10 Exercícios ... 571

Capítulo 25 Manutenção e Ambiente 573

25.1 Fundamentos sobre manutenção ... 573
25.2 Contratos de manutenção ... 574
Manutenção local ... 574
Manutenção para troca de placas ... 574
Garantias ... 574
25.3 Cuidados na manipulação de placas ... 574
Eletricidade estática ... 575
Reassentando placas ... 575
25.4 Monitores ... 575
25.5 Módulos de memória ... 575
25.6 Manutenção preventiva ... 576
25.7 Ambiente ... 576
Temperatura ... 576
Umidade ... 577
Refrigeração do escritório ... 577
Refrigeração de salas de máquinas ... 577
Monitoramento de temperatura ... 578
25.8 Potência ... 578

Sn❄w666

Controle remoto de potência ... 579
25.9 Racks ... 579
25.10 Ferramentas ... 579
25.11 Exercícios ... 580

Capítulo 26 Análise de Desempenho 581

26.1 O que você pode fazer para melhorar o desempenho ... 582
26.2 Fatores que afetam o desempenho ... 582
26.3 Verificação de desempenho do sistema ... 583
Analisando o emprego de CPU ... 583
Como o Linux gerencia a memória ... 585
Analisando o emprego de memória ... 586
Analisando o I/O de disco ... 588
26.4 Socorro! meu sistema simplesmente está uma lesma! ... 589
26.5 Leitura recomendada ... 590
26.6 Exercícios ... 590

Capítulo 27 Cooperando com o Windows 592

27.1 Compartilhamento de arquivos e de impressoras ... 592
NFS: Network File System ... 592
CIFS ... 593
SMBFS: monte compartilhamentos CIFS remotos ... 593
Samba: servidor CIFS para Linux ... 593
Instalando e configurando o Samba ... 594
Depurando o Samba ... 595
Autenticação unificada de logins no Windows e Linux ... 596
27.2 Emulação de terminais segura com o SSH ... 596
27.3 Emuladores x Windows ... 597
27.4 Clientes de e-mail em PCs ... 597
27.5 Backups em PCs ... 598
27.6 Inicialização dual ... 598
Montando sistemas de arquivos externos com o Linux ... 598
27.7 Executando aplicativos Windows sob o Linux ... 598
Linux (e UNIX) no Windows ... 599
27.8 Dicas de hardware ... 599
27.9 Leitura recomendada ... 600
27.10 Exercícios ... 600

Capítulo 28 Daemons 601

28.1 init: o processo primordial ... 602
28.2 cron e atd: comandos de agendamento ... 602
28.3 inetd e xinetd: daemons de gerenciamento ... 602
Configurando inetd ... 603
Configurando xinetd ... 604
O arquivo services ... 606
portmap/rpcbind: associa serviços RPC a portas TCP e UDP ... 606
28.4 Daemons de kernel ... 606
klogd: lê mensagens de kernel ... 607
28.5 Daemons de serviços de arquivos ... 607
rpc.nfsd: serve arquivos ... 607
28.6 Daemons de bancos de dados administrativos ... 608
ypbind: localiza servidores NIS ... 608
ypserv: servidor NIS ... 608
rpc.ypxfrd: transfere bancos de dados NIS ... 608

Snow666

nscd: atribuir nome ao daemon de cache de serviços ... 608
28.7 Daemons Internet ... 608
28.8 Daemons para sincronização de tempo ... 611
timed: sincroniza relógios ... 611
ntpd e xntpd: sincroniza ainda melhor os relógios ... 611
28.9 Daemons de inicialização e de configuração ... 611
28.10 Exercícios ... 612

Capítulo 29 Políticas e Diretrizes 613

29.1 A cultura Linux ... 614
Linux de Prateleira ... 614
Projetos Linux ... 615
29.2 Políticas e procedimento ... 615
Políticas de segurança ... 617
Acordos referentes a políticas para usuários ... 618
Acordos referentes a políticas para administradores de sistemas ... 619
Política e procedimentos para situações emergenciais ... 620
Planejamento em relação a desastres ... 620
Informações variadas adicionais ... 621
29.3 Questões legais ... 622
Responsabilidades ... 622
Criptografia ... 622
Direitos autorais ... 623
Privacidade ... 624
Impingindo uma política ... 624
Licenças de software ... 625
Spam: e-mail comercial não solicitado ... 626
Pesquisas de administradores de sistemas ... 627
Pesquisa salarial feita pela SAGE ... 627
Pesquisa salarial elaborada pelo SANS ... 628
29.4 Escopo do serviço ... 628
29.5 Sistemas de relatos de problemas ... 629
29.6 Gerenciando a gerência ... 629
29.7 Contratação, demissão e treinamento ... 630
Adequação de atitudes ... 631
Guerras entre operadores ... 631
Refinamento iterativo ... 631
29.8 Histórias de guerra e ética ... 632
Erro de chefe #1 ... 632
Erro de chefe #2 ... 632
Quem despedir? ... 633
Engenheiros *versus* Departamento de TI ... 633
Intruso Joe ... 633
Convites de casamento ... 634
Imagens GIF pornográficas ... 634
Migração de dados ... 634
Bush tem de morrer! ... 635
Histórias de segunda mão do World Trade Center ... 635
29.9 Documentatação local ... 636
29.10 Compras ... 637
29.11 Afastando hardware de serviço ... 637
29.12 Organizações, conferências e outros recursos ... 638
LPI: Linux Professional Institute ... 639
SAGE: System Administrators Guild ... 639
SANS: System Administration, Networking, and Security Institute ... 640

www.snow666.blogspot.com
Snow666

Listas de discussão e recursos Web 640
Material impresso 640
29.13 Padrões 640
LSB: Linux Standard Base 641
POSIX 641
29.14 Modelos de documentos 641
29.15 Leitura recomendada 642
29.16 Exercícios 642

Índice **643**

www.snow666.blogspot.com
Snow666

Prefácio

Estava bastante ansioso por ver em primeira mão esta obra somente para Linux, semelhante ao *UNIX System Administration Handbook*. A terceira edição deste último incluía uma cobertura sobre o Red Hat Linux, mas era apenas uma das quatro variantes distintas do UNIX. Esta versão do livro abrange as principais distribuições Linux e omite a maior parte do material que não é relevante ao Linux; estava curioso para ver o nível de diferença que isso produziria.

E, depois de ler, vi que produziu uma grande diferença. Este foi o resultado. As distribuições Linux partem de um conjunto comum de software de fonte aberto e, portanto, elas são mais similares entre si do que acontece com outras versões do UNIX. Conseqüentemente, o texto parece ter se tornado muito mais específico. Em vez de sugerir várias maneiras pelas quais o seu sistema *deveria* se comportar, os autores podem informá-lo exatamente como ele *efetivamente* se comporta.

Ao mesmo tempo, fica claro que toda a riqueza e variedade do software UNIX ainda está representada aqui. Simplesmente todos os softwares mais populares do mundo rodam no Linux nos dias de hoje, e as instalações Linux estão se defrontando com cada vez menos comprometimentos. À medida que fornecedores de peso como IBM, Oracle e Silicon Graphics adotam o Linux, ele está rapidamente se tornando o padrão universal com o qual as demais versões do UNIX estão sendo comparadas (e, nem sempre, favoravelmente!).

Como este próprio livro demonstra, os sistemas Linux são tão funcionais, seguros e confiáveis quanto seus equivalentes proprietários. Graças ao esforço contínuo de seus milhares de desenvolvedores, o Linux está mais pronto do que nunca para emprego das linhas de frente de seus milhares de desenvolvedores do mundo real. Os autores deste livro conhecem bem o terreno e estou feliz por deixá-lo em suas mãos capazes. Aproveite!

Linus Torvalds

www.sn0w666.blogspot.co
Sn❄w666

Página em branco

www.snow666.blogspot.com
Snow666

Introdução

O Linux é um sistema operacional relativamente novo no mundo da computação. Nascido no início dos anos 90, ele teve imensa publicidade e apoio da comunidade do software de fonte aberto. De várias maneiras, o Linux se tornou o representante antimatéria de um outro universo centrado na Microsoft.

Apesar das várias conquistas do Linux, ele ainda tem de ganhar total aceitação no mundo da "computação em ambiente de produção". Uma vez sinônimo de mainframes de peso, este ambiente é um universo no qual alguns poucos minutos podem representar perda de milhões de dólares, dezenas de empregos ou, em casos extremos, de vidas.

Pensamos que já estaria na hora do Linux ter sido aceito como um membro totalmente ordenado desta comunidade. Entretanto, tal aceitação pode se desenvolver apenas com a ajuda de uma cavalaria de administradores de sistema Linux *profissionais*.

Decidimos escrever um livro que viesse a se tornar o melhor amigo do administrador de sistemas Linux profissional. Onde apropriado, adaptamos os já provados conceitos e materiais de nosso popular livro, *UNIX System Administration Handbook*. Acrescentamos uma grande quantidade de material específico ao Linux e atualizamos o restante, mas grande parte da matéria abrangida continua a mesma. Esperamos que você concorde que o resultado é um guia de alta qualidade para administração de sistemas Linux que se beneficia de uma experiência anterior.

Há outros livros sobre administração de sistemas Linux, mas nenhum deles fornece a profundidade e a abrangência do material necessárias para efetivamente usar o Linux em ambientes comerciais do mundo real. Eis as características que diferenciam nosso livro:

- Adotamos uma postura prática. Nosso objetivo não é o de reafirmar o conteúdo de seus manuais, mas sim o de sintetizar nossa experiência coletiva em administração de sistemas. Este livro contém numerosas histórias de guerra e uma abundância de conselhos pragmáticos.
- Este livro não é sobre como rodar o Linux em casa, em sua garagem ou em seu PDA. Descrevemos o emprego do Linux em ambientes de produção como empresas, órgãos governamentais e universidades.
- Abrangemos as redes Linux com detalhes. Esse é o aspecto mais difícil da administração em rede e a área onde acreditamos poder ser de maior valia.
- Não simplificamos o material ao extremo. Nossos exemplos refletem situações da vida prática com todas as suas imperfeições e complicações não previstas. Na maioria dos casos, os exemplos foram extraídos diretamente de sistemas de produção.
- Abrangemos três distribuições principais do Linux.

AS DISTRIBUIÇÕES USADAS EM NOSSOS EXEMPLOS

Como tantos outros sistemas operacionais, o Linux cresceu e se ramificou em diversas direções. Embora o desenvolvimento do kernel tenha permanecido surpreendentemente centralizado, o empacotamento e a distribuição de sistemas operacionais Linux completos são menosprezados por uma série de grupos, cada qual com sua própria agenda.

Analisamos três distribuições em detalhe:

- Red Hat 7.2
- SuSE 7.3
- Debian 3.0

Optamos por essas distribuições porque se encontram entre as mais populares e porque são representativas da comunidade Linux como um todo. Entretanto, grande parte do material apresentado neste livro também se aplica a outras distribuições.

Snow666

Damos informações detalhadas sobre cada uma dessas distribuições para cada tópico que discutimos. Comentários específicos a um determinado sistema operacional são marcados com o logotipo da distribuição.

A Organização Deste Livro

Este livro é dividido em três grandes partes: Administração Básica, Redes e Um Bocado de Coisas.

A parte Administração Básica fornece uma visão geral do Linux sob a perspectiva de um administrador de sistemas. Os capítulos dessa parte cobrem a maior parte dos dados e técnicas necessários para rodar um sistema Linux dedicado.

Redes descreve os protocolos usados em sistemas Linux e as técnicas empregadas para configurar, ampliar e manter redes. Também é visto software de rede de alto nível. Entre os tópicos apresentados estão o DNS (Domain Name System, sistema de nomes de domínios), o NFS (Network File System, sistema de arquivos de rede), roteamento, **sendmail** e gerenciamento de redes.

A parte Um Bocado de Coisas inclui uma série de informações suplementares. Alguns capítulos discutem pacotes de software opcional como o sistema de impressão Linux; outros dão conselhos sobre tópicos que vão desde manutenção de hardware até as políticas para operação de uma instalação Linux.

Cada capítulo é seguido por um conjunto de exercícios práticos. Os itens são classificados por nossa estimativa de empenho necessário para completá-los, em que "empenho" é um indicador tanto da dificuldade da tarefa quanto do tempo necessário para completá-la.

Há quatro níveis:

sem estrelas	Fáceis, devem ser imediatos
★	Mais difíceis ou mais longos, podem exigir trabalho experimental
★★	O mais difícil ou mais longo, exigem trabalho experimental e investigação profunda
★★★★★	Projetos de duração semestral (somente em alguns poucos capítulos)

Alguns dos exercícios requerem acesso root ou **sudo** ao sistema; outros exigem a permissão do grupo de administradores de sistema local. Ambas as exigências são mencionadas no texto do exercício.

Nossos Contribuidores

Ficamos encantados com a participação de Adam Boggs, Matt Crosby e Ned McClain como autores-colaboradores. O conhecimento profundo desses profissionais numa série de áreas enriqueceu em muito o conteúdo deste livro. Devemos a eles agradecimentos especiais por tornarem possível a realização desta obra. Adam fez um trabalho maravilhoso, fornecendo mais do que havia prometido. Matt foi um mestre em tirar um coelho da cartola e Ned foi o nosso sempre entusiasta e tão necessário "pau para toda obra".

Informações para Contato

Fizemos todo o possível para que este livro não tivesse erros, mas sabemos que não somos infalíveis. Se você encontrar algum erro, ou mesmo quiser enviar sugestões ou comentários, envie um e-mail para clientes@pearsoned.com.

Se deseja contato direto com os autores, envie um e-mail (em inglês) para sa-book@admin.com. Os autores costumam responder à maioria dos e-mails, porém você tem de ser paciente; algumas vezes demora um pouco até que um deles possa responder. Para obter uma cópia atualizada da lista de bugs dos autores e outras informações de última hora, visite o site www.admin.com.

Esperamos que você goste deste livro e desejamos muita sorte em suas aventuras no mundo da administração de sistemas!

Evi Nemeth
Garth Snyder
Trent R. Hein

www.snow666.blogspot.com
Snow666

Agradecimentos

Muitas pessoas auxiliaram na concretização deste livro, de uma maneira ou de outra, ajudando em tudo, desde revisões técnicas ou sugestões de exercícios a apoio moral. Estas pessoas merecem agradecimentos especiais por estarem sempre presentes:

Eric Allman
Piet Barber
Jon Corbet
Jim Dennis
Barbara Dijker
Bill Fenner
Robert Gray
Andreas Gustafsson
Cricket Liu
Paul Lussier
Derek Martin
Lynda McGingley
Laszlo Nemeth
Gretchen Phillips
Josh Prismon
Greg Shapiro
Paul Vixie

Mary Franz, nossa editora, merece não somente nossos agradecimentos especiais como também um prêmio por lidar com sucesso de autores intratáveis (um bando deles nos vem à mente). Mary foi infinitamente paciente conosco, mesmo em ocasiões em que não merecíamos este tratamento, e ela fez de tudo para nos encorajar a trilhar um processo de concentração na busca de qualidade.

Mary Lou Nohr mais uma vez realizou um trabalho extraordinário como editora de texto. Ela é uma demolidora e uma flor, ambas em uma só pessoa.

Gostaríamos de agradecer a Hank Pantier e Jeff Lindermeier da Hungry Mind Interactive Technologies por nos ajudarem a atualizar a capa em formato de desenho animado cada vez mais eclética. Robert Gray, David Clements e Cyrus Hall da Boulder Labs nos auxiliaram nos exercícios. Mais uma vez, Pat Parseghian veio em nosso socorro nos oferecendo apoio moral e disposição para ser nosso "agente especial". Obrigado, Pat!

O departamento de Ciências da Computação da Universidade do Colorado merece agradecimentos especiais por nos ter oferecido recursos computacionais e numerosos "estudos de caso". O departamento de Ciências da Computação da Brown University em Providence, RI, forneceu temporariamente a Evi com um local para trabalho com acesso à Internet enquanto ela tentava conciliar as atividades deste livro com o trabalho em seu veleiro.

www.sn0w666.blogspot.co
Sn❄w666

Página em branco

www.snow666.blogspot.com
Snow666

Sobre os Autores

Evi Nemeth se aposentou da Faculdade de Ciências da Computação da Universidade de Colorado, mas ainda "brinca" na pesquisa de redes na CAIDA (Cooperative Association for Internet Data Analysis) no San Diego Supercomputer Center. Atualmente, ela está explorando o Caribe em seu novo brinquedo, um veleiro de 40 pés chamado Wonderland.

evi@cs.colorado.edu

Garth Snyder trabalhou na NeXT e na Sun e é graduado em Engenharia Elétrica pelo Swarthmore College. No momento é candidato a MD/MBA na Universidade de Rochester.

snyderga@simon.rochester.edu

Trent R. Hein é co-fundador da Applied Trust Engineering, uma empresa que presta serviços de consultoria sobre segurança e desempenho em infra-estruturas de redes. Trent possui o título de bacharel em Ciências da Computação pela Universidade de Colorado.

trent@atrust.com

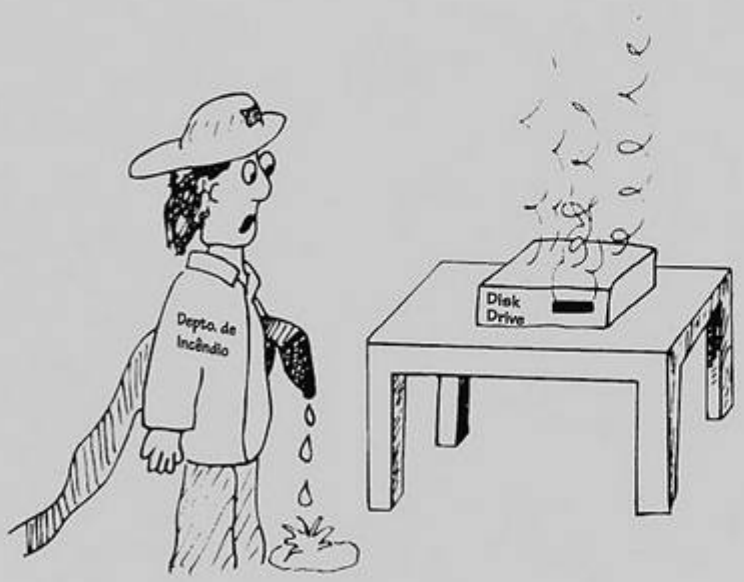
www.sn0w666.blogspot.co
Sn❄w666

Página em branco

Snow666

SEÇÃO 1

ADMINISTRAÇÃO BÁSICA

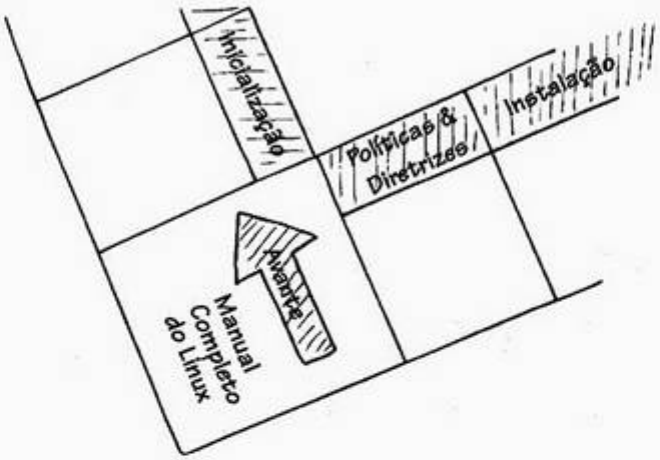


www.sn0w666.blogspot.co
Sn❄w666

Página em branco

Snow666

1 Por Onde Começar



Decidimos escrever um livro que seja um verdadeiro companheiro do administrador de sistemas, fornecendo conselhos práticos, conforto e teoria básica da administração de sistemas que não se pode extrair da leitura de manuais. Como resultado, este livro destina-se a complementar — não substituir — a documentação do Linux.

Acreditamos que ele o ajudará em cinco aspectos:

- Dará uma visão geral dos principais sistemas administrativos, identificando as diferentes partes de cada um deles e explicando como elas operam em conjunto.
- Introduzirá técnicas administrativas gerais que acreditamos, pela experiência prática, que valem a pena.
- Ajudará a escolher soluções que continuam a funcionar bem mesmo quando sua instalação aumenta em tamanho e complexidade.
- Orientará a separar as boas das más idéias e o educará em termos de diversas barbaridades cometidas pelos distribuidores.
- Resumirá procedimentos comuns de forma que não será necessário, no caso da execução de tarefas simples, você ficar vasculhando no meio daquela montanha de detalhes contidos em manuais.

É impossível executar essas funções com objetividade perfeita, porém, acreditamos que deixamos relativamente claras ao longo do texto quais são nossas tendências. Uma das coisas interessantes em relação à administração de sistemas é o fato de pessoas razoáveis terem noções radicalmente diferentes de quais são as políticas e procedimentos mais apropriados. Apresentamos a você nossas opiniões subjetivas como se fossem matéria-prima. Ficará a seu cargo decidir o quanto aceitar e em que extensão aplicar nossos comentários à sua instalação.

1.1 Background Sugerido

Partimos do princípio, neste livro, que você já tenha uma certa experiência em Linux ou UNIX. Em particular, você deve ter uma noção geral de como o Linux se apresenta sob a perspectiva do usuário antes de mergulhar na administração. Vários livros podem ajudá-lo a acelerar este processo; veja a lista apresentada na página 14.

A maioria das tarefas administrativas é executada editando arquivos de configuração e escrevendo scripts, de modo que você deve estar familiarizado com um editor de texto. Para o desalento de muitos, usar o Microsoft Word como único editor de texto é um obstáculo significativo para uma administração de sistemas eficiente.

www.snow666.blogspot.com
Snow666

Recomendamos enfaticamente que você aprenda a usar o **vi**. Ele é um padrão em todos os sistemas Linux e, embora possa parecer um tanto fraco quando comparado a opções mais ostensivas como **emacs**, ele é perfeitamente utilizável. Também gostamos do **pico**, que é um editor simples e de baixo impacto para administradores de sistemas novatos. Ele é incluído em várias distribuições. Seja cauteloso em relação a editores não padronizados; se você se viciar no uso de um deles, acabará se cansando de levá-lo consigo para instalá-lo em cada sistema novo.

Um dos pilares da administração de sistema (e um tema que é abordado ao longo do livro) é o emprego de scripts para automatizar tarefas administrativas. Para ser um administrador eficiente, você tem de ser capaz de ler e modificar scripts **sh** (que no mundo Linux são, na verdade, scripts **bash**). Os scripts que criamos da estaca zero podem ser escritos na linguagem de scripting ou shell de sua preferência.

Para novos projetos de scripting, recomendamos Perl. Como linguagem de programação, ela é um pouco estranha (está bem, mais do que um pouco). Entretanto, inclui efetivamente muitos recursos indispensáveis para os administradores de sistemas. O livro editado pela O'Reilly, *Programming Perl*, de Larry Wall et al., é um texto-padrão; ele também é um modelo para escrever bem assuntos técnicos. Uma relação completa é dada na página 14.

Verifique em cpan.org para ter uma lista completa de software Perl útil.

Muitos administradores preferem Python em vez de Perl e sabemos de instalações que estão fazendo um esforço concentrado para conversão de Perl para Python. Python é uma linguagem "mais agradável" do que Perl e, geralmente, os scripts em Python são mais legíveis e de manutenção mais fácil. Entretanto, Python não possui a grande base de usuários e a inércia de Perl; você terá de julgar quais recursos são mais importantes para a sua instalação. Um conjunto de links úteis comparando Python a outras linguagens de scripting (inclusive Perl) pode ser encontrado em `http://www.python.org/doc/Comparisons.html`

1.2 O Relacionamento do Linux com o UNIX

Alguns poderão chamar isto de heresia, porém é um ponto que deve ser enfatizado ao extremo: o Linux é uma versão do UNIX. Ele é uma reimplementação de baixo para cima do padrão POSIX, roda em várias plataformas de hardware distintas e é compatível com a maior parte do software UNIX existente. Ele difere da maior parte das demais versões do UNIX no fato de ser gratuito, com código-fonte aberto e desenvolvido de maneira cooperativa, com colaborações provenientes de centenas de indivíduos e organizações.

A comunidade Linux às vezes minimiza a importância do relacionamento do Linux com o UNIX e enfatiza sua independência em termos de desenvolvimento. O Linux realmente *é* diferente do UNIX em sua implementação e em seu gerenciamento técnico e é compreensível que os desenvolvedores Linux gostariam de ser reconhecidos como algo mais do que simplesmente clonadores do UNIX. Não obstante, as diferenças entre o UNIX e o Linux não são importantes sob a perspectiva dos usuários e administradores de sistema. De fato, os sistemas Linux geralmente se comportam mais como sistemas UNIX tradicionais do que as variantes excêntricas do UNIX como o Solaris da Sun ou o AIX da IBM.

O software Linux *é* software UNIX. Graças, em grande parte, ao Projeto GNU, a maior parte do software importante que dá aos sistemas UNIX seu valor foi desenvolvida sob alguma forma de modelo de código-fonte aberto. O mesmo código roda em sistemas Linux e não-Linux. Por exemplo, o servidor Web Apache não se importa realmente se está rodando Linux ou HP-UX. Sob o ponto de vista de aplicativos proprietários, o Linux é simplesmente uma das variantes do UNIX melhor suportadas.

Também vale a pena notar que o Linux não é a única versão gratuita do UNIX existente no mundo. FreeBSD, NetBSD e OpenBSD, todas ramificações do Berkley Software Distribution da UC Berkley, possuem seguidores ardorosos. Esses sistemas operacionais são geralmente comparáveis ao Linux em termos de recursos e confiabilidade, embora eles tenham um pouco menos de suporte de terceiros (fornecedores de software).

Os sistemas UNIX vêm sendo usados em ambientes de produção há vários anos. Agora, o Linux amadureceu a ponto de, também, ser um sistema operacional em nível de ambiente de produção. Este livro, diferentemente da maioria dos demais sobre administração de sistemas Linux, concentra-se no emprego efetivo do Linux num ambiente de produção — não simplesmente como um desktop de um único usuário.

1.3 Histórico do Linux e do UNIX

O Linux se originou em 1991 como um projeto pessoal de Linus Torvalds, um universitário finlandês. Ele concebeu o projeto originalmente como uma modesta ramificação do Minix, um sistema operacional modelo escrito por Andrew S. Tanenbaum. Entretanto, o Linux gerou grande interesse no mundo como um todo, e o kernel rapidamente assumiu vida própria. Explorando o poder do desenvolvimento corporativo, Linus foi capaz de empreender uma tarefa muito mais ambiciosa. O kernel versão 1.0 foi lançado em 1994; na época em que a primeira edição deste livro estava sendo escrita (início de 2002), a versão estável mais recente do kernel do Linux era 2.4.17.

www.snow666.blogspot.com
Snow666

Pelo fato de o Linux possuir grande parte das características de seus ancestrais UNIX, não é muito razoável apontar a origem da era Linux para o ano de 1991. A história do UNIX vem de várias décadas atrás (1969) quando o UNIX nasceu como um projeto de pesquisa no AT&T Bell Labs. Em 1976 o UNIX tornou-se disponível gratuitamente nas universidades e, portanto, tornou-se a base de várias classes de sistemas operacionais e de projetos de pesquisa acadêmicos.

O Berkley UNIX começou em 1977 quando o CSRG (Computer Systems Research Group) na Universidade da Califórnia, Berkeley, licenciou código da AT&T. As versões do Berkeley (chamadas BSD, de Berkeley Software Distribution) começaram em 1977 com o 1BSD para o PDP-11 e culminaram em 1993 com o 4.4 BSD.

À medida que o UNIX ganhava aceitação comercial, o preço das licenças dos fontes aumentou rapidamente. Eventualmente, Berkeley estabeleceu o objetivo a longo prazo de eliminar o código pertencente à AT&T do BSD, um processo tedioso e demorado. Antes que o trabalho pudesse ser completado, Berkeley perdeu as verbas para a pesquisa de sistemas operacionais e o CSRG foi abandonado.

Antes de sucumbir, o CSRG lançou o seu conjunto final de código isento de código da AT&T, conhecido como 4.4 BSD-Lite. A maioria das versões atuais do BSD UNIX (inclusive o BSD/OS, FreeBSD, NetBSD e OpenBSD) reclama o pacote 4.4 BSD-Lite como sendo seu antepassado.

Grande parte das demais versões do UNIX (incluindo HP-UX, Solaris e IRIX) é descendente da linhagem AT&T original. Os sistemas Linux enquadram-se numa situação mais ou menos intermediária.

1.4 Distribuições Linux

O projeto do Linux difere de outras variações do UNIX no aspecto em que ele define somente um kernel de sistema operacional. O kernel tem de ser empacotado juntamente com comandos, daemons e outros softwares para formar um sistema operacional completo e utilizável — em termos de Linux, uma "distribuição". Todas as distribuições do Linux compartilham da mesma linhagem de kernel, porém os materiais auxiliares que vêm junto com este kernel podem variar razoavelmente entre estas distribuições.

As distribuições variam em termos de foco, suporte e popularidade. A Tabela 1.1 lista as distribuições de propósito genérico mais populares.

Tabela 1.1 As distribuições Linux de propósito genérico mais populares[a,b].

Distribuição	Site Web	Comentários
Conectiva	www.conectiva.com	Distribuição de propósito genérico, originária e forte no Brasil; multilíngüe.
Debian	www.debian.org	Associado mais de perto ao projeto GNU.
Linux Mandrake	www.mandrake.com	Afirma ser refinado, porém repleto de recursos.
Red Hat Linux	www.redhat.com	A distribuição de propósito genérico mais largamente utilizada.
Slackware Linux	www.slackware.com	Antigamente, um Linux dominante, hoje em dia menos popular.
SuSE Linux	www.suse.com	Particularmente forte na Europa; multilíngüe.
TurboLinux	www.turbolinux.com	Forte na Ásia; suporte a idiomas asiáticos.

a. As distribuições são listadas em ordem alfabética, não em ordem de preferência ou de popularidade.
b. Além desses, outro de destaque era a distribuição Caldera que foi suspensa, como está informado no site distrowatch.com — consta uma carta aos clientes enviada pela SCO, adquirente da Caldera, na qual a SCO notifica a suspensão das atividades relativas ao Linux e especificamente ao seu produto OpenLinux.

Várias distribuições menores não são apresentadas na Tabela 1.1 e muitas distribuições não listadas se destinam a grupos com necessidades específicas (como, por exemplo, desenvolvedores de sistemas embutidos). Um guia mais abrangente para as distribuições, inclusive aquelas que não se encontram na língua inglesa, pode ser encontrado em www.linux.org/dist.

Portanto, qual é a melhor distribuição?

Uma rápida pesquisa na Internet revelará que isto é uma das coisas mais freqüentemente perguntadas — e pelo menos, freqüentemente respondidas — sobre o Linux. A resposta correta para o seu caso depende de como você pretende usar o sistema, as variedades de UNIX com as quais você já está familiarizado, suas preferências políticas e suas necessidades de suporte.

Snow666

A maioria das distribuições do Linux pode fazer praticamente tudo o que você tenha imaginado com um sistema Linux. Algumas delas podem requerer a instalação de software adicional para serem totalmente funcionais e outras podem facilitar determinadas tarefas; entretanto, suas diferenças não são muito significativas. Na realidade, há um certo mistério no porquê de existirem tantas distribuições diferentes, cada uma das quais clamando "instalação fácil" e uma "gigantesca biblioteca de programas" como característica de distinção. É difícil evitar a conclusão de que as pessoas simplesmente gostam de criar distribuições do Linux.

Por outro lado, já que o foco deste livro está no gerenciamento de instalações Linux de grande porte, somos parciais com distribuições como o Red Hat que levam em consideração o gerenciamento de redes de máquinas. Algumas distribuições são desenvolvidas tendo em mente ambientes de produção e outras não. As migalhas extras de assistência que os sistemas orientados a ambientes de produção fornecem podem provocar uma diferença significativa na facilidade em que os sistemas são administrados.

Ao adotar uma distribuição, você está investindo na maneira de fazer as coisas de um determinado fornecedor. Em vez de olhar somente os recursos do software instalado é melhor considerar como a sua empresa e aquele fornecedor em questão irão trabalhar um com o outro nos anos vindouros. Algumas perguntas importantes a serem feitas são as seguintes:

- Esta distribuição ainda existirá daqui a cinco anos?
- Esta distribuição estará no topo dos patches de segurança mais recentes?
- Esta distribuição lançará software atualizado prontamente?
- Se tiver problemas, o fornecedor conversará com você?

Analisadas sob este ângulo, algumas das pequenas distribuições mais interessantes e não convencionais não parecem muito convidativas. Por outro lado, as distribuições mais viáveis não são necessariamente as mais indicadas para um ambiente corporativo. Por exemplo, acreditamos que o Debian Linux (tá bom, tá bom, o Debian GNU/Linux!) permaneça viável por um bom tempo apesar do fato da Debian não ser uma empresa; não vender nada e não oferecer nenhum tipo de suporte.

A LinuxPlanet mantém uma lista de distribuições do Linux em www.linuxplanet.com, juntamente com uma série de análises profundas. Algumas dessas análises são um tanto fora de época, mas elas ainda dão uma boa idéia das características de cada distribuição. Também há alguns comentários relativamente claros e não tendenciosos sobre as principiais distribuições em www.linux.com; veja *Introduction to Linux and Linux.com*.

Neste livro, usamos três distribuições populares do Linux como exemplos: o Red Hat Linux 9, o SuSE Linux 8.2 e o Debian GNU/Linux 3.0r1. Esses sistemas representam uma fatia do mercado Linux corporativo e contabilizam a maioria das instalações Linux em uso hoje em dia em grandes sistemas.

Ferramentas administrativas específicas a cada distribuição

Muitas distribuições fornecem ferramentas orientadas visualmente (como o Red Hat Network Administration Tool ou o YaST do SuSE) que o ajudam a configurar ou administrar aspectos selecionados do sistema. Estas ferramentas podem ser muito úteis, especialmente para os administradores novatos, mas elas efetivamente tendem a obscurecer os detalhes daquilo que está ocorrendo quando você faz modificações. Neste livro, abordamos os mecanismos subjacentes aos quais as ferramentas gráficas se referem em vez de abordarmos as próprias ferramentas, por várias razões.

A primeira delas: as ferramentas visuais tendem a ser proprietárias ou pelo menos específicas a determinada distribuição — elas introduzem variação nos processos que podem vir efetivamente a ser bem inconsistentes entre as distribuições num nível mais baixo. Em segundo lugar, acreditamos que é importante para os administradores terem uma compreensão exata de como os seus sistemas funcionam. Quando um sistema pifa, normalmente as ferramentas visuais não são úteis na localização e solução de problemas. Finalmente, a configuração manual em geral é evidentemente melhor: é mais rápida, mais flexível e mais confiável e também mais fácil de ser colocada em scripts.

1.5 Convenções Tipográficas e de Notação

Neste livro, os nomes de arquivos, comandos e argumentos literais para comandos são apresentados em negrito. Substitutos (isto é, argumentos de comandos que não devem ser interpretados literalmente) se encontram em itálico. Por exemplo, no comando

cp *file directory*

supõe-se que você substituirá *file* e *directory* pelos nomes de um arquivo e um diretório realmente existentes.

www.snow666.blogspot.com
Snow666

Trechos de arquivos de configuração e sessões em terminais são mostrados numa fonte de largura fixa[1]. Algumas vezes, indicamos sessões interativas com texto em itálico. Por exemplo:

```
% grep Bob /pub/phonelist        /* Procura o numero de telefone do Bob */
Bob Knowles 555-2834
Bob Smith 555-2311
```

Afora esses casos específicos, tentamos manter num mínimo possível as fontes especiais e convenções de formatação desde que não comprometêssemos a inteligibilidade. Por exemplo, muitas vezes falamos de entidades como o grupo Linux chamado daemon e o printer anchor-lw, sem nenhuma formatação especial.

Em geral, usamos as mesmas convenções adotadas nos manuais para indicar a sintaxe de comandos:

- Qualquer coisa entre colchetes ("[" e "]") é opcional.
- Qualquer coisa seguida de reticências ("...") pode ser repetida.
- Chaves ("{" e "}") significam que deveríamos selecionar um dos itens separados por barras verticais "|").

Por exemplo, a especificação:

```
bork [-x] (on|off) nomedoarquivo ...
```

serviria para um dos seguintes comandos:

```
bork on /etc/passwd
bork -x off /etc/passwd /etc/termcap
bork off /usr/lib/tmac
```

Usamos os caracteres globais estilo shell para determinados padrões de busca:

- Um asterisco (*) coincide com zero ou mais caracteres.
- Um ponto de interrogação (?) coincide com um caractere.
- Um til (~) representa o diretório inicial do usuário atual.
- *~user* representa o diretório inicial de *user*.

Por exemplo, poderíamos nos referir aos diretórios de scripts de inicialização Debian **/etc/rc0.d**, **/etc/rc1.d** e assim por diante com o padrão abreviado **/etc/rc*.d**.

O texto dentro das aspas normalmente tem um significado técnico preciso. Nesses casos, ignoramos as regras normais do inglês e colocamos a pontuação fora das aspas de forma que não possa haver nenhuma dúvida sobre o que está incluso ou não.

Informações específicas de sistema

As informações contidas neste livro geralmente se aplicam a todas as distribuições de exemplo a menos que seja dada uma atribuição específica. Detalhes particulares a uma distribuição são marcados com o logotipo do fornecedor:

Red Hat Linux 97.2

SuSE Linux 8.2

Debian GNU/Linux 3.0r1

Esses logotipos são usados com o tipo de permissão de seus respectivos proprietários. Entretanto, os distribuidores não revisaram nem endossaram o conteúdo deste livro.

1.6 Onde Buscar Informações

A documentação do Linux é exibida numa série de fontes, algumas das quais você encontrará instaladas em seu sistema e outras que residem na Internet. As mais importantes são:

1. Na verdade, não é realmente uma fonte de largura fixa, mas ela se parece com uma. Nós gostamos mais desta do que as fontes de largura fixa que experimentamos. É por isso que as colunas em alguns exemplos podem parecer não se alinharem perfeitamente.

www.snow666.blogspot.com
Snow666

- Páginas de manual (man pages), lidas por meio do comando **man**
- Documentos Texinfo, lidos por meio do comando **info**
- HOWTOs, pequenas notas sobre vários assuntos (www.linuxdoc.org)
- Guias, tratados mais longos sobre vários assuntos (www.linuxdoc.org)
- Documentação específica a uma determinada distribuição

As páginas de manual e os documentos Texinfo constituem a documentação "on-line" tradicional (embora, obviamente, toda a documentação esteja on-line de uma forma ou de outra). Esses documentos são tipicamente instalados com o sistema; páginas de manual específicas a um determinado programa normalmente vêm de carona sempre que você instala um sistema.

As páginas de manual são descrições concisas de comandos individuais, drivers, formatos de arquivos ou rotinas de biblioteca. Elas não cobrem tópicos mais gerais do tipo "Como instalo um novo dispositivo?" ou "Por que o meu sistema é tão lento?". Para essas perguntas, você deve consultar os HOWTOs.

Os documentos Texinfo foram criados há muito tempo pelo pessoal do GNU em reação ao fato de o comando **nroff** usado para formatar páginas de manual ser proprietário da AT&T. Nos dias de hoje temos o próprio **groff** do GNU para executar essa tarefa para nós, e a questão do **nroff** já não é mais importante. Infelizmente, muitos pacotes GNU persistem em documentar-se com arquivos Texinfo em vez de páginas de manual. Além de definirem um segundo padrão desnecessário de documentação, o Texinfo é um pequeno sistema de hipertexto que por si só é mais complicado do que um labirinto. **info info** irá iniciá-lo nos obscuros mistérios do Texinfo.

Felizmente, os pacotes documentados com Texinfo normalmente instalam stubs páginas de manual que o informam como usar o comando **info** para ler informações de determinados pacotes. Você pode aderir com segurança ao comando **man** para realizar buscas manuais e mergulhar no mundo **info** somente quando for instruído para tal.

HOWTOs e guias são mantidos pelo Linux Documentation Project, acessível on-line em www.linuxdoc.org. O LDP é um repositório central para todos os tipos de informações úteis sobre o Linux. Ele também concentra esforços para traduzir documentos relativos ao Linux em outras linguagens.

Alguns guias LDP on-line gratuitos, em inglês, de particular relevância para os administradores de sistema são o *The Linux System Administrators' Guide* de Lars Wirzenius, Joanna Oja e Stephen Stafford; o *Advanced Bash-Scripting Guide* de Mendel Cooper; *The Linux Network Administrator's Guide*, 2.ª edição, de Olaf Kirch e Terry Dawson; e *Linux System Administration Made Easy*, de Steve Frampton.

Infelizmente, muitos dos documentos LDP não são mantidas de forma contínua. Já que os anos-Linux são muito parecidos com anos de amizade em sua relação com o tempo real, documentos abandonados (sem acompanhamento constante) são candidatos a rapidamente tornarem-se obsoletos. Sempre verifique a data num HOWTO ou guia e pese sua credibilidade correspondente.

Muitas das partes mais importantes da base de software Linux são mantidas por terceiras partes neutras como o Internet Software Consortium e a Apache Software Foundation. Esses grupos normalmente fornecem documentação adequada para os pacotes por eles distribuídos. As distribuições algumas vezes empacotam o software mas restringem na documentação, de modo que normalmente é útil verificar no fonte original para ver se não há materiais adicionais disponíveis.

Uma outra fonte útil de informações sobre o projeto de muitos pacotes de software Linux é a série de documentos "Request for Comments" que descreve os protocolos e procedimentos usados na Internet. Verifique a página 137 para obter mais informações.

Organização das páginas de manual

As páginas de manual do Linux são divididas tipicamente em nove seções conforme ilustrado na Tabela 1.2.

Algumas seções são subdivididas mais ainda. Por exemplo, a seção 3M contém páginas de manual para a biblioteca matemática do sistema. As seções 6 e 9 são normalmente vazias. Muitos sistemas possuem uma seção de manuais chamada "l" para páginas de manual locais. Uma outra convenção comum é a seção "n" para subcomandos específicos a software (como os agregados **bash**).

www.snow666.blogspot.com
Snow666

Tabela 1.2 Seções das páginas de manual do Linux.

Seção	Conteúdo
1	Aplicativos e comandos em nível de usuário.
2	Chamadas de sistema e códigos de erros de kernel.
3	Chamadas de biblioteca.
4	Drivers de dispositivos e protocolos de redes.
5	Formatos de arquivos padrão.
6	Jogos e demonstrativos.
7	Arquivos e documentos variados.
8	Comandos de administração de sistema.
9	Interfaces e especificações de kernel obscuras.

A entrada **nroff** para páginas de manual normalmente é armazenada nos diretórios **/usr/share/man/man***X*, onde X é um dígito de **1** a **9** ou **l** ou **n**. As páginas são normalmente compactadas através do **gzip** para poupar espaço. (O comando **man** sabe como descompactá-las durante a execução.) Versões formatadas dos manuais são mantidas em **/var/cache/man/cat***X*. O comando **man** formatará as páginas de manual conforme a necessidade; se os diretórios **cat** forem graváveis, **man** também depositará páginas formatadas à medida que forem criadas, gerando um cache de páginas de manual comumente lidas.

O comando **man** na verdade pesquisa uma série de diretórios diferentes para localizar páginas de manual que você solicita. Podemos determinar o caminho de busca por meio do comando **manpath**. Esse caminho (do Red Hat) é típico:

```
% manpath
/usr/share/man:/usr/man:/usr/X11R6/man:/usr/kerberos/man:/usr/local/man
```

Se necessário, podemos configurar nossa variável de ambiente MANPATH para prevalecer em relação ao caminho-padrão. Também configuramos o padrão abrangendo o sistema em **/etc/man.config** (Red Hat) ou **/etc/manpath.config** (Debian e SuSE).

man:read manual pages

man *title* formata uma página de manual específica e a envia para o seu terminal com **less** (ou seja lá qual for o programa especificado em sua variável de ambiente PAGER), *title* normalmente é um comando, dispositivo ou nome de arquivo. As seções do manual são pesquisadas grosseiramente em ordem numérica, embora as seções que descrevam comandos (seções 1, 8 e 6) normalmente sejam procuradas em primeiro lugar.

A forma **man** *section title* obtém uma página de manual de uma seção específica. Portanto, **man tty** lhe dá a página de manual para o comando **tty** e **man 4 tty** lhe apresenta a página de manual correspondente ao driver de terminal controlador.

man -k *keyword* imprime uma lista de páginas de manuais que possuem *keyword* (palavra-chave) em suas sinopses de uma linha. Por exemplo:

```
% man -k translate
mhbuild (1)    - translate MIME composition draft
objcopy (1)    - copy and translate object files
pfbtops (1)    - translate a PostScript font in .pfb format to ASCII
tr (1)         - translate on delete characters
...
```

Outras fontes de informações sobre o Linux

Há um grande mundo adorável lá fora. Possivelmente não seríamos capazes de mencionar cada conjunto útil de informações sobre Linux, ou até mesmo somente os principais, porém algumas fontes significativas são apresentadas na Tabela 1.3.

Snow666

Tabela 1.3 Recursos Linux na Web.

Site Web	Descrição
www.linux.com	Depósito central de informações sobre o Linux (não oficial).
www.linux.org	Um outro depósito central de informações sobre o Linux (não oficial).
www.linuxdoc.org	Linux Documentation Project (Projeto de Documentação sobre o Linux).
www.linuxtoday.com	Revista on-line Linux Today.
www.linuxplanet.com	Mais uma revista on-line, bom conteúdo para administração de sistemas.
www.freshmeat.net	Catálogo enorme de software para Linux e UNIX.
www.linuxhq.com	Compilação de correções e informações relacionadas a kernel.
www.kernel.org	Site oficial sobre kernel Linux.
www.linuxapps.com	Biblioteca de software Linux.
www.tucows.com	Biblioteca de software multiplataforma com conteúdo Linux.
www.linuxworld.com	Revista on-line da Computerworld.

Não fique envergonhado de também acessar recursos de UNIX gerais — grande parte dessas informações se aplicam diretamente ao Linux. Uma grande quantidade de informações sobre administração de sistemas se encontra disponível na Internet, sob várias formas. Podemos, por exemplo, digitar sysadmin questions em qualquer dos populares mecanismos de busca como www.google.com, www.yahoo.com, www.altavista.com e www.webopedia.com. Uma lista de outros recursos "de partida" podem ser encontrada no Capítulo 29, *Políticas e Diretrizes*.

Muitos sites suprem diretamente as necessidades dos administradores de sistema. Eis abaixo alguns dos quais gostamos especialmente:

- www.ugu.com — UNIX Guru Service; um bocado de informações para administradores de sistema.
- www.stolely.com — um belo conjunto de links de recursos para administradores de sistemas.
- www.tucows.com — software para Windows e Macintosh, filtrados por qualidade.
- slashdot.org — "O ponto" das notícias para geeks.
- www.cpan.org — uma central de bibliotecas e scripts Perl.
- securityfocus.com — informações relacionadas com segurança de sistemas; banco de dados imenso e pesquisável sobre vulnerabilidades em sistemas.

Um outro recurso útil e divertido é a página "Roseta Stone" de Bruce Hamilton no endereço:

`http://bhami.com/rosetta.html`

Ele fornece ponteiros para os comandos e ferramentas usadas para várias tarefas de administração em vários sistemas operacionais.

1.7 Como Localizar e Instalar Software

A maioria das distribuições do Linux divide seu software em pacotes que podem ser instalados de maneira independente um do outro. Ao instalar o Linux num computador novo, normalmente selecionamos um conjunto de pacotes "iniciais" para serem copiados para o novo sistema.

Essa arquitetura simplifica muitos aspectos da configuração de um sistema e é uma das principais vantagens do Linux em relação a versões tradicionais do UNIX. Infelizmente, esse design também complica a tarefa de escrever sobre tais distribuições, pois nunca fica realmente claro quais pacotes "fazem parte de" uma dada distribuição. Um pacote será "incluído" se ele estiver nos CDs de instalação, mas não fizer parte da instalação-padrão? Somente se ele estiver em cada computador que estiver rodando esta distribuição? Se ele se encontrar nos CDs "bônus" estes virão somente com a versão completa da distribuição?

Neste livro, geralmente descrevemos a instalação-padrão, isso não significa necessariamente que o pacote não estará em *seu* sistema ou que ele não é suportado pela sua distribuição. Eis como descobrir se você o tem e, em caso negativo, como obtê-lo.

Primeiramente, use o comando **which** do shell para descobrir se um comando relevante já se encontra em seu caminho de pesquisa. Por exemplo, o comando a seguir revela que o servidor Web Apache já foi instalado nesta máquina em **/usr/sbin**:

Snow666


```
% which httpd
/usr/sbin/httpd
```

Se **which** não puder localizar o comando que procura, tente **whereis**, que pesquisa um intervalo mais amplo de diretórios de sistema e é independente do caminho de pesquisa de seu shell. Tome cuidado também que o comando **which** de alguns sistemas não mostrará arquivos para os quais você não tem permissão para executar. Por exemplo:

```
% which ippd
/usr/bin/which: no ippd in (/bin:/usr/bin:/sbin:/usr/bin)
% whereis ippd
ippd: /usr/sbin/ippd
% ls -l /usr/sbin/ippd
-rwx------      1 root     root      124924 Aug 3 2000 /usr/sbin/ippd
```

Uma outra alternativa é o comando incrivelmente útil chamado **locate**, que consulta um índice pré-compilado do sistema de arquivos para localizar nomes de arquivos que atendem a um determinado padrão de busca. Ele não é específico a comandos ou pacotes, porém pode localizar qualquer tipo de arquivo. Por exemplo, se não estiver certo onde encontrar o arquivo include **signal.h** (que é a fonte oficial para as definições de signal do Linux), você poderia tentar:

```
% locate signal.h
/usr/include/asm/signal.h
/usr/include/linux/signal.h
/usr/include/signal.h
/usr/include/sys/signal.h
```

O banco de dados de **locate** normalmente é restaurado toda noite por meio do comando **updatedb**, que é executado a partir do **cron**. Conseqüentemente, os resultados de um **locate** nem sempre refletem mudanças recentes do sistema de arquivos.

Caso você saiba o nome de um pacote que está procurando, também poderá usar os utilitários de empacotamento de seu sistema para verificar diretamente a presença do pacote. Por exemplo, num sistema Red Hat ou SuSE, o comando a seguir verifica a presença da linguagem de scripting Python:

```
% rpm -q python
python-1.5.2-27
```

Veja o Capítulo 23, *Instalação e Localização de Software*, para obter maiores informações sobre os comandos de empacotamento dos exemplos de sistema apresentados.

Se o pacote em que estiver interessado parecer não estar instalado, pode ser mais fácil procurá-lo na Internet em vez de retornar aos seus CDs de instalação originais. Você também terá mais chances de acabar encontrando uma versão atualizada do pacote, já que muitos são atualizados com freqüência.

A maior parte do software Linux é desenvolvida por grupos independentes que lançam o software na forma de código-fonte. Os distribuidores Linux pegam então o código-fonte, o compilam de maneira apropriada para as convenções em uso em seus sistemas em particular e empacotam os binários resultantes. Normalmente é mais fácil instalar um pacote binário específico a uma determinada distribuição do que recuperar e compilar o código-fonte original. Entretanto, os distribuidores normalmente estão uma ou duas versões atrás da versão atual.

O fato de duas distribuições usarem o mesmo sistema de empacotamento não significa necessariamente que os pacotes para os dois sistemas sejam intercambiáveis. Por exemplo, tanto o Red Hat quanto o SuSE usam RPM, porém os layouts de seus sistemas de arquivos são ligeiramente diferentes. É sempre melhor usar pacotes desenvolvidos para a sua distribuição em particular caso eles estejam disponíveis.

A primeira parada em sua pesquisa de software geralmente deve ser o site Web de seu distribuidor. A maioria das distribuições fornece um banco de dados de pacotes pesquisável na Web juntamente com uma rede de espelhos FTP por meio dos quais o download dos pacotes em si pode ser feito. Se tudo o mais falhar, tente procurar o pacote num site de downloads como o freshmeat.net ou efetuar uma busca Google com o nome do pacote desejado.

1.8 Tarefas Essenciais de um Administrador de Sistemas

As seções a seguir dão uma visão geral de algumas tarefas que se espera que os administradores de sistema realizem. Estas atribuições não precisam necessariamente ser executadas por apenas uma pessoa, e em várias instalações o trabalho é dividido entre vários profissionais. Porém, realmente há necessidade de pelo menos uma pessoa que entenda de todas estas tarefas e que garanta que alguém as faça.

Snow666

Veja o Capítulo 6 para obter maiores informações sobre como adicionar novos usuários.

Adicionando e removendo usuários

O administrador de sistemas acrescenta contas para novos usuários e elimina contas de usuários que não estão mais ativos. O processo de adicionar e remover usuários pode ser automatizado, mas certas decisões administrativas (onde colocar o diretório inicial do usuário, em quais máquinas criar a conta etc.) têm de ser feitas antes de um novo usuário poder ser acrescentado.

Quando um usuário não tiver mais de ter acesso ao sistema, a conta deste deve ser desabilitada. Todos os arquivos de propriedade dessa conta têm de ser gravados em fita e depois deletados, para que o sistema não acumule espaço indesejado ao longo do tempo.

Veja os Capítulos 8, 12 e 24 para obter maiores informações sobre estes tópicos.

Adicionando e removendo hardware

Quando um novo hardware é comprado ou quando um hardware é transferido de uma máquina para outra, o sistema tem de ser configurado para reconhecer e usar esse hardware. As tarefas de suporte de hardware podem variar da simples adição de uma impressora até atribuições mais complexas como adicionar uma unidade de disco.

Veja o Capítulo 10 para obter maiores informações sobre backups.

Realizando backups

Realizar backups é talvez a tarefa mais importante do administrador de sistemas e também é normalmente a mais ignorada ou feita de maneira descuidada. Os backups são enfadonhos e tomam muito tempo, porém eles são absolutamente necessários. Os backups podem ser automatizados e delegados a um subordinado, contudo ainda é tarefa do administrador de sistemas certificar-se de que os backups sejam executados corretamente e dentro do cronograma (e de que a mídia resultante possa ser realmente usada para restaurar arquivos).

Instalando software novo

Quando um novo software é adquirido, ele deve ser instalado e testado, normalmente em diversos tipos de hardware e sistemas operacionais. Assim que o software estiver funcionando corretamente, os usuários têm de ser informados de sua disponibilidade e localização. Software local deve ser instalado num lugar que seja fácil de diferenciar do software de sistema. Essa organização simplifica a tarefa de atualizar o sistema operacional, já que o software local não será sobregravado pelo procedimento de atualização.

Monitorando o sistema

Grandes instalações exigem supervisão permanente. Dentre essas atividades diárias temos de garantir que o serviço Web e de e-mail esteja funcionando corretamente, verificar arquivos de log para reconhecer antecipadamente sinais de problemas, garantir que as redes locais estejam todas apropriadamente conectadas e ficar de olho na disponibilidade dos recursos de sistema como, por exemplo, espaço em disco.

Diagnosticando

Os sistemas Linux e o hardware no qual são executados ocasionalmente falham. É tarefa do administrador fazer o papel de mecânico, diagnosticando problemas e utilizando os serviços de experts caso necessário.

Normalmente é mais difícil diagnosticar o problema do que resolvê-lo.

Verifique a página 639 para obter sugestões referentes à documentação.

Mantendo documentação local

À medida que um sistema é modificado para atender às necessidades de uma empresa, ele começa a diferir do sistema comum descrito pela documentação. É obrigação do administrador de sistemas documentar aspectos do sistema específicos ao ambiente local. Essa tarefa inclui documentar qualquer software que é instalado mas não vem com o sistema operacional, documentar por onde os cabos passam e como eles são construídos, manter os registros de manutenção para todo o hardware, registrar o estado dos backups e documentar políticas e procedimentos locais.

Verifique o Capítulo 21 para obter maiores informações sobre segurança.

Auditando a segurança

O administrador de sistemas deve implementar uma política de segurança e verificar periodicamente para garantir que a segurança do sistema não foi violada. Em sistemas de baixa segurança, esta tarefa poderia envolver apenas algumas checagens superficiais em

Snow666

termos de acesso não autorizado. Num sistema de alta segurança, isso pode incluir uma elaborada rede de pontos de controle e programas de auditoria.

Ajudando os usuários

Embora ajudar os usuários quanto a seus diversos problemas raramente esteja incluído na descrição das tarefas pertinentes ao cargo de administrador de sistemas, isso consome uma parte significativa do dia de trabalho da maioria dos administradores. Os administradores de sistemas são bombardeados com problemas que vão desde "Meu programa funcionou ontem e agora não! O que você modificou?" até "Eu derramei café no meu teclado! Devo jogar água nele para deixá-lo bem limpinho?".

1.9 Administração de Sistemas sob Pressão

Os administradores de sistemas atuam em várias posições. No mundo real, eles normalmente são pessoas com outros serviços às quais foi pedido que dessem uma olhadinha em alguns computadores. Se estiver nessa situação, talvez queira pensar um pouquinho a respeito de onde isso poderia eventualmente levá-lo.

Quanto mais você aprende sobre seu sistema, mais a comunidade de usuários irá depender de você. As redes invariavelmente crescem e talvez você seja pressionado a gastar uma parte maior de seu tempo na administração do sistema. Logo você perceberá que é a única pessoa dentro da empresa que sabe como executar uma série de tarefas importantes.

Assim que os companheiros de trabalho começarem a vê-lo como o administrador do sistema local, dificilmente você conseguirá se livrar desta incumbência. Já que muitas tarefas administrativas são intangíveis, você também descobrirá que os outros já o vêem como administrador em tempo integral, bem como um engenheiro de sistemas, escritor ou secretária em tempo integral.

Alguns administradores de má vontade tentam agir defensivamente em relação às solicitações adotando uma atitude geniosa e fornecendo serviços de baixa qualidade. Não recomendamos essa postura; ela faz você ser malvisto e cria problemas adicionais.

Em vez disso, sugerimos que você documente o tempo que gastou na administração do sistema. Nosso objetivo deve ser o de manter o trabalho num nível controlável e juntar evidências para poderem ser usadas no momento em que você solicitar uma diminuição na carga de tarefas administrativas. Na maioria das empresas, você precisará fazer lobby com a gerência durante um período de seis meses a um ano para ser substituído; portanto, planeje com antecedência.

Por outro lado, talvez você descubra que gosta da área de administração de sistemas e que anseia se tornar um administrador de sistemas em tempo integral. Suas perspectivas para o cargo são boas. Infelizmente, seus problemas políticos provavelmente aumentarão. Verifique o Capítulo 29, *Políticas e Diretrizes*, para uma pré-estréia dos horrores que o aguardam.

A Síndrome de Personalidade dos Administradores de Sistemas

Uma condição clínica infeliz, porém comum, resultante do trabalho como administrador de sistemas é a Síndrome de Personalidade dos Adminstradores de Sistema. O início desta condição normalmente aparece logo no princípio do terceiro ano da carreira de um administrador de sistemas e a síndrome pode durar até ele chegar à sua aposentadoria. Entre os sintomas característicos estão, mas não se limita apenas a:

- Pagerfobia fantasmagórica aguda: o sentimento perturbador de que seu pager pifou (quando na realidade isso não aconteceu) e que sua noite tranqüila com a sua cara-metade está prestes a ser interrompida abruptamente, resultando numa maratona de trabalho de 72 horas sem comida.
- Vodugrafia de usuários: a criação compulsiva de representações de bonecos-vodu do subconjunto da sua população de usuários que parece não entender que a persistente falta de planejamento por parte deles não constitui uma emergência em seu mundo.
- Fitaplexia anal idiopática: ímpeto repentino tarde da noite de montar fitas de backup para verificar se elas são realmente legíveis e foram etiquetadas corretamente.
- Inaplicabilidade científica: o forte desejo de sacudir violentamente outros administradores de sistema que parecem jamais ter encontrado o método científico.

Podem ser usadas várias terapias curativas para tratar essa triste condição. As mais eficientes são desenvolver um bom senso de humor e a construção de uma pequena, porém bem provisionada, adega. Talvez você também precise considerar a abordagem mais meditativa de fitar um ponto qualquer no espaço e juntar os calcanhares sempre que as palavras "Deu pau no servidor de novo?" forem proferidas nas vizinhanças. Se tudo isso falhar, tire umas férias.

Snow666

1.10 Leitura Recomendada

Embora haja muitos livros introdutórios sobre o Linux no mercado, descobrimos que a maioria deles é muito ..., bem, ruim. Em geral, é melhor você ficar com os "clássicos" do UNIX. Quase tudo o que você ler a respeito do UNIX se aplicará igualmente ao Linux.

ANDERSON, GAIL; ANDERSON, PAUL. *The UNIX C Shell Field Guide*. Englewood Cliffs, NJ: Prentice Hall. 1986
LAMB, LINDA; ROBBINS, ARNOLD. *Learning the vi Editor, 6.ª edição*. Sebastopol, CA: O'Reilly and Associates. 1998.
ABRAHAMS, PAUL W.; LARSON, BRUCE A. *UNIX for the Impatient*, 2.ª edição. Reading, MA: Addison-Wesley. 1995.
PEEEK, JERRY; O'REILLY, TIM; LOUKIDES, MIKE. *UNIX Power Tools, 2.ª edição*. Sebastopol, CA: O'Reilly & Associates. 1997.
MONTGOMERY, JOHN. *The Underground Guide to UNIX: Sligthly Askew Advice from Unix Guru*. Reading, MA: Addison-Wesley. 1995.
REICHARD, KEVIN; FOSTER-JOHNSON, Eric. *Unix in Plain English, 3.ª edição*. Foster City, CA: IDG Books Worldwide. 1999.
WALL, LARRY; CHRISTIANSEN; TOM, ORWANT; JON. *Programming Perl, 3.ª edição*. Sebastopol, CA: O'Reilly & Associates. 2000.
CHRISTIANSEN, TOM; TORKINGTON,NATHAN; WALL, LARRY. *Perl Cookbook*. Sebastopol, CA: O'Reilly & Associates. 1998.
WELSH, MATT; KALLE DALHEIMER, MATHIAS;, KAUFMAN, Lar. *Running Linux, 3.ª edição*. Sebastopol, CA: O'Reilly & Associates. 1999.
GANCARZ, MIKE. *The UNIX Philosophy*. Woburn, MA: Digital Press, 1994.

1.11 Exercícios

E1.1 Que comando você usaria para ler sobre a chamada de sistema **sync** (*não* o comando *sync*)? Como você leria uma página de manual referente a **sync** mantida em **/usr/local/share/man**?

E1.2 Há um arquivo de configuração abrangente do sistema que controle o comportamento de **man** em sua instalação? Que linhas você acrescentaria a esse arquivo caso quisesse armazenar documentação local em **/doc/man**? Que estrutura de diretórios você teria de usar em **/doc/man** para torná-lo membro da hierarquia de página de manual?

E1.3 Quais são as principais diferenças entre **man** e **info**? Quais são algumas vantagens de cada um deles?

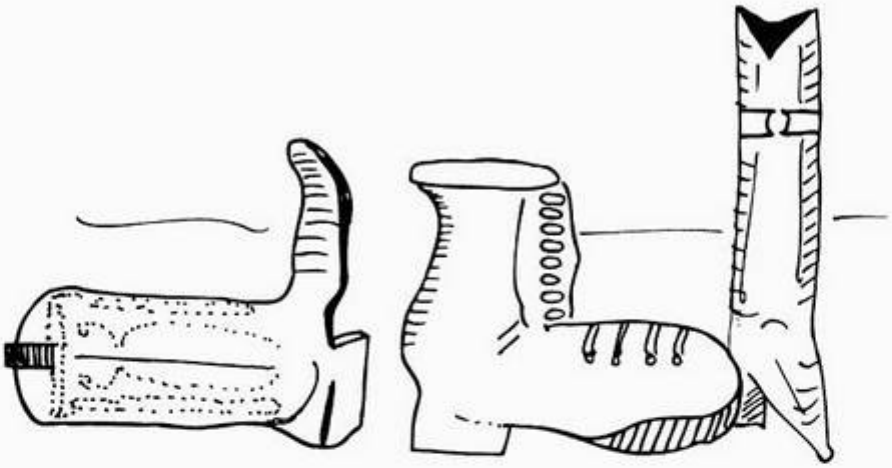
★ **E1.4** Qual é o estágio atual de desenvolvimento do Linux? Quais são alguns dos principais participantes desse mercado? Como o projeto é gerenciado?

★ **E1.5** Pesquise várias distribuições do Linux (use a página 5 como ponto de partida) e recomende uma distribuição para cada um dos seguintes aplicativos. Explique os motivos de sua escolha:
a) Um único usuário trabalhando num home office
b) O laboratório de computação de uma universidade
c) Um servidor Web corporativo

★ **E1.6** Suponha que você descubra que um certo recurso do **httpd** Apache parece não funcionar conforme documentado no Red Hat 7.2.
a) O que você deve fazer antes de relatar isto como um bug?
b) Caso decida que o bug seja real, quem você deve notificar e de que maneira?
c) Quais informações têm de ser incluídas para tornar útil um relatório de bug?

www.snow666.blogspot.com
Snow666

2 Inicialização e Desligamento



O Linux é um sistema operacional complexo; ligar e desligar sistemas Linux é mais complicado do que simplesmente abaixar a chave geral. Ambas as operações têm de ser realizadas corretamente caso se queira que o sistema permaneça saudável.

O processo de bootstrapping é similar ao desenvolvimento de relacionamentos entre seres humanos. Ele varia muito, conforme determinado pelos anos precoces de desenvolvimento e é considerado pela maioria como uma "questão privada". Olhar o processo como um observador pode ocasionalmente ser grotesco e evocar sentimentos de impotência e de desânimo.

Embora este capítulo surja logo no início do livro, ele se refere a assuntos que não são discutidos detalhadamente até chegarmos a muitas centenas de páginas adiante. Em particular, a familiaridade com o material do Capítulo 5, *O Sistema de Arquivos*, do Capítulo 12, *Drivers e o Kernel*, e do Capítulo 28, *Daemons*, será útil. Se seu sistema já inicializar sem qualquer problema, talvez você queira pular este capítulo e retornar a ele mais tarde.

Uma restrição adicional: o processo de inicialização (booting) é dependente do hardware. As informações que se seguem são genericamente verdadeiras, mas podem diferir da realidade do seu sistema.

2.1 Bootstrapping

Bootstrapping é o termo nerd equivalente a "inicializar um computador". Os recursos normais fornecidos pelo sistema operacional não se encontram disponíveis durante o processo de inicialização de modo que o computador tem de "se virar sozinho". Durante o bootstraping, o kernel é carregado na memória e começa a ser executado. São executadas uma série de tarefas e o sistema é então tornado disponível aos usuários.

O tempo de boot é um período de vulnerabilidade especial. Erros em arquivos de configuração, equipamentos faltantes ou não confiáveis e sistemas de arquivos danificados podem impedir que um computador entre no ar. A configuração de boot normalmente é uma das primeiras tarefas que um administrador tem de realizar num sistema novo. Infelizmente, ela também é uma das mais difíceis e requer alguma familiaridade com muitos outros aspectos do Linux.

Quando um computador é ligado, ele executa o código de boot que se encontra armazenado em ROM. Esse código tenta descobrir como carregar e iniciar o seu kernel. O kernel sonda o hardware do sistema e depois gera o processo **init** do sistema que sempre é PID 1.

Várias coisas têm de acontecer antes de um prompt de login poder aparecer na tela. Os sistemas de arquivos têm de ser verificados e montados e os daemons de sistema inicializados. Esses procedimentos são gerenciados por uma série de scripts shell que são executados em seqüência por **init**. Os scripts de inicialização são normalmente re-

Snow666

feridos como "arquivos rc" devido à maneira como são chamados; "rc" vem de "runcom" ou "run command", um remanescente histórico do sistema operacional CTSS, por volta de 1965. O layout exato dos scripts de inicialização e a maneira pela qual eles são executados varia entre os diferentes sistemas. Veremos os detalhes posteriormente, ainda neste capítulo.

Booting automático e manual

Os sistemas Linux podem ser inicializados tanto no modo automático como no manual. No modo automático, o sistema executa o procedimento de inicialização completo por sua conta, sem qualquer ajuda externa. No modo manual, o sistema segue o procedimento automático até um determinado ponto para então passar o controle a um operador antes da maioria dos scripts de inicialização ter sido exectuada. Nesse ponto, o computador se encontra em "modo monousuário". A maioria dos processos de sistema não está sendo executada e outros usuários não poderão entrar no sistema.

Na operação do dia-a-dia, o booting automático é usado quase exclusivamente. Um procedimento de inicialização típico para uma máquina moderna é o de um usuário ligar o computador e aguardar (e aguardar ...) até que o sistema entre no ar. Não obstante, é importante entender o procedimento de boot automático e saber como executar um boot manual. Normalmente, você terá de fazer um boot manual quando algum problema interferir no boot automático, por exemplo, no caso de um sistema de arquivos corrompido ou de uma interface de rede configurada de forma inadequada.

Etapas no processo de inicialização

Um processo de bootstrapping típico consiste em seis etapas distintas:

- Carga e inicialização do kernel
- Detecção e configuração de dispositivos
- Criação de processos de sistema espontâneos
- Intervenção do operador (somente no boot manual)
- Execução dos scripts de inicialização do sistema
- Operação multiusuário

Os administradores têm pouco controle sobre a maioria dessas etapas. Efetuamos a maior parte da configuração de bootstrap editando os scripts de inicialização do sistema.

Veja o Capítulo 12 para obter maiores informações sobre o kernel.

Inicialização do kernel

O kernel Linux é por si só um programa, e a primeira tarefa de bootstrapping é obter este programa na memória de forma que ele possa ser executado. O nome do caminho do kernel usualmente é **/vmlinuz** ou **/boot/vmlinuz**.

O Linux implementa um processo de carga de dois estágios. Durante o primeiro estágio, a ROM do sistema carrega na memória um pequeno programa de boot proveniente do disco. Esse programa organiza para que o kernel seja carregado. Tal procedimento ocorre fora do domínio do Linux e, portanto, não é padronizado entre os diferentes sistemas.

O kernel efetua testes de memória para descobrir quanta memória RAM está disponível. Algumas estruturas de dados internas do kernel são dimensionadas estaticamente, para que o kernel separe uma quantidade fixa de memória real para si mesmo quando ele é inicializado. Essa memória é reservada para o kernel e não pode ser usada por processos em nível de usuario. O kernel imprime na console uma mensagem informando a quantidade total de memória física e a quantidade disponível para processos de usuário.

Configuração de hardware

Uma das primeiras tarefas do kernel é verificar o ambiente da máquina para ver que hardware está presente. Ao construir um kernel para seu sisetma, você o informa sobre que dispositivos de hardware ele deve esperar encontrar; quando o kernel começa a executar, ele tenta localizar e inicializar cada dispositivo sobre os quais você o informou. O kernel imprime uma linha de informações "criptográficas" para cada dispositivo que ele encontra.

As informações de dispositivo fornecidas em tempo de configuração de kernel em geral não são especificadas completamente. Nesses casos o kernel tenta determinar as demais informações de que ele precisa sondando o barramento em busca de dispositivos e buscando informações nos drivers apropriados. Os drivers para dispositivos que estão faltando ou que não respondem a uma sondagem serão desabilitados. Se um dispositivo for conectado posteriormente

Snow666

ao sistema, também pode ser possível carregar ou habilitar um driver para ele durante a execução. Verifique o Capítulo 12, *Drivers e o Kernel*, para maiores detalhes.

Processos de sistema

Assim que tiver terminado a inicialização básica, o kernel cria vários processos "espontâneos" no espaço de usuário. Eles são chamados processos espontâneos porque não são criados através do mecanismo de sistema normal (**fork**); verifique a página 40 para obter maiores detalhes.

O número e a natureza de processos espontâneos varia de sistema para sistema. Sob o Linux, não há nenhum PID 0 visível. **init** (sempre processo 1) é acompanhado por vários processos em memória e de manipulador de kernel, entre os quais **kflushd**, **kupdate**, **kpiod** e **kswapd**. De todos esses, somente **init** é realmente um processo de usuário totalmente preparado. Os demais são, na verdade, partes do kernel que foram arrumados para se parecerem com processos por razões de escalonamento ou de arquitetura.

Assim que os processos espontâneos tiverem sido criados, o papel do kernel no bootstrapping estará completado. Entretanto, nenhum dos processos que manipulam operações básicas (como aceitação de logins) foi criado, nem a maioria dos daemons Linux for inicializada. Todas essas tarefas são cuidadas por **init** (indiretamente, em alguns casos)

Intervenção do operador (somente para boot manual)

Se o sistema tem de ser ativado no modo monousuário, um flag de linha de comando passado pelo kernel notifica **init** deste fato quando ele inicia. **init** eventualmente passa o controle para **sulogin**, uma versão especial assexuada, porém extremamente radical, de **login** que solicita a senha root[1]. Se você introduzir a senha correta, o sistema irá gerar um shell root. Você pode dar um <Control-D>[2] em vez de digitar a senha para evitar o modo monousuário e continuar no modo multiusuário.

Veja o Capítulo 3 para obter mais informações sobre a conta root.

Por meio do shell monousuário, podemos executar comandos quase da mesma forma que ao estarmos logados num sistema completamente inicializado. Entretanto, em sistemas SuSE e Debian, em geral, somente a partição-raiz é montada; temos de montar manualmente outros sistemas de arquivos para usar programas que não residem em **/bin**, **/sbin** ou **/etc**. Daemons normalmente não rodam no modo monousuário; desse modo, comandos que dependem de processos de servidor (por exemplo, **mail**) não funcionarão corretamente.

O comando **fsck** é normalmente executado durante uma inicialização automática para verificar e reparar sistemas de arquivos. Ao ativar o sistema em modo monousuário, talvez seja preciso executar **fsck** manualmente. Veja a página 109 para maiores informações sobre o **fsck**.

Quando o shell monousuário é abandonado, o sistema tenta continuar inicializando no modo multiusuário.

Execução de scripts de inicialização

No momento em que o sistema estiver pronto para executar seus scripts de inicialização, ele é reconhecidamente Linux. Muito embora ele ainda não pareça com um sistema totalmente inicializado, não há mais nenhuma etapa "mágica" no processo de inicialização. Os scripts de iniciailização são simplesmente scripts shell comuns e são selecionados e executados por **init** de acordo com um algoritmo que, embora algumas vezes tortuoso, é relativamente compreensível.

O cuidado, nutrição e a taxonomia dos scripts de inicialização merecem uma seção própria. Isso é contado com mais detalhes um pouco mais à frente.

Veja a página 85 para obter mais informações sobre o processo de login.

Operação multiusuário

Depois de os scripts de inicialização terem sido executados, o sistema estará totalmente operacional, exceto pelo fato de que ninguém estará apto a fazer o login. Para os logins serem aceitos num determinado terminal (inclusive a console), um processo **getty** tem de estar escutando-o. **init** gera diretamente estes processos **getty**, completando o processo de inicialização. **init** também é responsável pela geração de sistemas de login gráficos como **xdm** ou **gdm,** caso o sistema esteja configurado para usá-los.

1. Na verdade, isso nem sempre é verdadeiro em sistemas Red Hat. Dependendo de as opções fornecidas para o carregador de boot, talvez lhe seja solicitado a senha root antes de entrar no modo monousuário.
2. O Linux usa <Control-D> como indicador padrão de final de arquivo (EOF), de forma que digitá-lo no shell monousuário significa que você acabou de executar os comandos de usuário.

www.snow666.blogspot.com
Snow666

Tenha em mente que **init** continua a desempenhar um importante papel mesmo após o bootstrapping estar completo. **init** possui um "nível de execução" monousuário e vários "níveis de execução" multiusuário que determinam quais dos recursos do sistema estão ativados. Os níveis de execução serão descritos posteriormente neste capítulo.

2.2 Inicializando PCs

Até agora vimos a descrição genérica do processo de inicialização. Agora revisitaremos várias das etapas mais importantes (e complicadas) para acrescentar alguns detalhes adicionais relevantes ao mundo dos PCs.

A inicialização de PCs é uma longa provação que requer uma boa dose de informações para ser explicada. Caso não precise lidar com PCs, você poderá pular diretamente para a página 21.

Em que aspectos um PC é diferente de hardware proprietário

Quando uma máquina é inicializada, ela começa executando código armazenado em ROMs. A localização e a natureza exata desse código varia, dependendo do tipo de máquina que você tenha. Numa máquina projetada explicitamente para UNIX ou um outro sistema operacional proprietário, o código está tipicamente em firmware, ou seja, ele sabe como usar os dispositivos conectados à máquina, como conversar com a rede num nível básico e como interpretar sistemas de arquivos baseados em disco. Tal firmware onisciente é muito conveniente para administradores de sistemas. Por exemplo, podemos simplesmente digitar o nome de arquivo de um novo kernel que o firmware saberá como localizar e ler este arquivo.

Em PCs, o código de boot inicial é geralmente chamado de BIOS (Basic Input/Output System, sistema básico de entrada/saída) e é extremamente simplista quando comparado ao firmware de uma máquina proprietária. Na verdade, um PC tem vários níveis de BIOS: um para a máquina em si, outro para a placa de vídeo e um para a placa SCSI se o sistema tiver uma instalada.

A BIOS embutida sabe da existência de alguns dispositivos que residem na placa-mãe, tipicamente a controladora IDE (e discos), teclado, portas seriais e paralelas. As placas SCSI normalmente não estão cientes dos dispositivos conectados a elas. Solucionar todos esses conflitos e interações entre todas as várias BIOS pode ser um verdadeiro pesadelo. Em muitos casos, uma confusão enorme reina sobre qual BIOS tem de escolher a partir de qual dispositivo deve fazer o boot.

O processo de inicialização de um PC

As BIOS modernas são um pouco mais espertas do que costumavam ser antigamente. Elas normalmente permitem que introduzamos um modo de configuração em tempo de boot pressionando-se e mantendo-se pressionadas uma ou duas teclas especiais; BIOS melhores até chegam à inovação de dizer-nos quais são essas teclas especiais, sem a necessidade de procurarmos por elas no manual.

A BIOS normalmente nos permitirá selecionar os dispositivos nos quais queremos tentar fazer o boot, o que soa mais promissor do que realmente acontece. Normalmente podemos especificar algo como: "Tente fazer o boot na unidade de disquete, depois tente fazer o boot no CD-ROM e, finalmente, no disco rígido". Infelizmente, a BIOS geralmente é limitada a fazer o boot na primeira unidade de CD-ROM IDE ou no primeiro disco rígido IDE. Se você foi muito, mas muito bonzinho mesmo no ano que passou, talvez Papai Noel lhe dê de presente uma BIOS que reconheça a presença de placas SCSI.

Gostaríamos de lhe dar alguma orientação útil para negociar essa encrenca toda. Infelizmente, isso é impossível; esta fase do processo de inicialização está totalmente sob controle dos fabricantes de hardware e suas miseráveis BIOS. Você terá de negociar com eles para ter ajuda.

A partir do momento em que a máquina tiver descoberto em qual dispositivo fazer o boot, ela tentará carregar os primeiros 512 bytes do disco. Esse segmento de 512 bytes é conhecido como MBR (Master Boot Record, registro mestre de boot). O MBR contém um programa que diz ao computador em qual partição de disco ele deve carregar um programa de boot secundário (o "boot loader"). Para maiores informações sobre partições de disco estilo PC e sobre o MBR, verifique o Capítulo 8, *Adicionando um Disco*.

O MBR-padrão é um programa simples que informa o computador para obter o seu boot loader na primeira partição no disco. O Linux fornece um MBR mais sofisticado que sabe como lidar com vários sistemas operacionais e kernels.

Assim que o MBR tiver escolhido a partição por meio da qual fazer o boot, ele tenta carregar o boot loader específico daquela partição. O boot loader fica então responsável pela carga do kernel.

Snow666

2.3 BOOT LOADERS: LILO E GRUB

O que seria da vida sem as opções? Nos dias de hoje podemos optar por um de dois boot loaders: LILO e GRUB. O LILO é o boot loader tradicional do Linux; ele é muito estável e bem documentado. O GRUB é o novo-rico recém-chegado; ele é mais flexível e também mais complexo.

LILO: o boot loader tradicional do Linux

O LILO vem com quase todas as distribuições do Linux, entre as quais Red Hat, SuSE e Debian. Os sistemas Red Hat agora dão opções de boot loaders durante o processo de instalação. Se estiver usando o LILO, os scripts de instalação baixarão uma cópia barata do LILO com opções genéricas; você terá muito pouco o que dizer neste processo.

O LILO pode ser instalado tanto no MBR do disco quanto no registro de boot da partição-raiz do Linux.

O LILO é configurado e instalado através do comando **lilo**. A configuração instalada pelo **lilo** toma como base o conteúdo do arquivo **/etc/lilo.conf**. Para mudar sua configuração de boot, simplesmente atualize o arquivo **/etc/lilo.conf** e execute novamente o comando **lilo**. O LILO tem de ser reconfigurado toda vez que o processo de boot muda — em particular toda vez que quisermos acrescentar uma nova partição de boot e toda vez que tivermos um novo kernel para ser inicializado.

Eis um arquivo **lilo.conf** básico para um sistema Linux que tem um kernel de produção, bem como um kernel de backup:

```
boot=/dev/hda            # Insere boot loader no MBR
root=/dev/hda1           # Especifica a partição-raiz
install=/boot/boot.b
map=/boot/map
delay=20                 # 2 segundos para interrupção do usuário
image=/vmlinuz           # Kernel a ser inicializado
   label=linux           # Label para se referir a esta entrada
   read-only

image=/vmlinuz-backup    # Entrada de backup
   label=backup
   read=only
```

Cada situação de boot possível tem um rótulo. No momento do boot, podemos informar ao LILO qual delas usar introduzindo o rótulo apropriado. O primeiro rótulo a aparecer em **lilo.conf** torna-se o padrão.

A situação-padrão (chamada `linux`) inicializa o arquivo **/vmlinuz**. A tag `read-only` especifica que o kernel deve montar o seu sistema de arquivos-raiz somente para leitura. Essa opção sempre deve estar presente; os scripts de inicialização cuidarão da remontagem da partição para leitura e escrita no momento adequado. Esse sistema também é configurado para inicializar um kernel de backup, **/vmlinuz-backup**. É sempre uma boa idéia ter uma alternativa como essa.

Executar o comando **lilo** sem argumentos gerará e instalará o boot loader e lhe informará que entradas se encontram disponíveis. Ele coloca uma estrela junto à imagem-padrão. Entretanto, se você tivesse cometido um erro no arquivo **lilo.conf**, o comando **lilo** normalmente não conseguiria descobrir o problema até chegar à metade do processo de instalação do boot loader. Quando isso acontece, o boot loader fica num estado confuso. *Não reinicialize* até ter completado o comando **lilo**. Para evitar entrar numa situação dessas, você pode executar **lilo -t** para testar a configuração sem realmente instalá-la. Se tudo parecer OK, você poderá então executar o comando **lilo** de verdade. É um tanto misterioso a razão pela qual o **lilo** não executa esse pré-teste para o usuário como situação-padrão.

A saída do comando **lilo** quando executada com o arquivo config acima é:

```
# lilo
Added linux*
Added backup
```

Quando o sistema é inicializado, LILO imprime o seguinte:

```
LILO:
```

Depois ele aguarda 2 segundos (20 décimos de segundo, configurado através da tag `delay`), depois inicializa o kernel **/vmlinuz** e monta a primeira partição do primeiro disco IDE como partição-raiz. Você pode ver uma lista de situações de inicialização definidas pressionando a tecla <Tab>:

Snow666


```
LILO: <Tab>
linux backup
LILO:
```

Para inicializar usando uma situação alternativa, simplesmente introduza seu rótulo no prompt.

GRUB: o loader GRand Unified Boot

GRUB é um substituto completo do LILO. É uma opção do mercado de reposição que pode ser acrescentada a qalquer sistema Linux; ele também recebe apoio oficial do Red Hat. O GRUB é particularmente popular entre os usuários que rodam uma variedade de sistemas operacionais (como Windows, OpenBSD, FreeBSD etc.) na mesma máquina ou que trabalham ativamente em desenvolvimento de kernels.

O GRUB também é útil para o pessoal que muda a configuração do sistema com freqüência. Diferentemente do LILO, que deve ser reinstalado no registro de boot ou MBR cada vez que você mudar o arquivo **lilo.conf**, o GRUB lê seu arquivo de configuração durante o boot, eliminando uma etapa extra.

Instalamos o GRUB num drive de boot executando **grub-install**. Esse comando usa como argumento o nome do dispositivo no qual se estará fazendo o boot. Infelizmente, o GRUB possui seu próprio modo de nomear dispositivos de disco físicos que é muito diferente da convenção Linux padrão. O nome de um dispostiivo GRUB é parecido com o seguinte:

```
(hd0,0)
```

O primeiro valor numérico indica o número do drive físico (começando de zero) e o segundo valor numérico representa o número da partição (repetindo, começando de zero). Neste exemplo (hd0,0) é equivalente ao dispositivo Linux **/dev/hda1**. Logo, caso queira instalar o GRUB em seu drive primário, você usaria o comando

```
grub-install '(hd0,0)'
```

As aspas são necessárias para impedir que o shell tente interpretar os parênteses de seu modo todo particular.

Por padrão, o GRUB lê sua configuração de inicialização padrão por meio de **/boot/grub/grub.conf**. Eis um exemplo de arquivo **grub.conf**:

```
default=0
timeout=0
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
title Red Hat Linux (2.4.7-10)
   root (hd0,0)
   kernel /boot/vmlinuz-2.4.7-10 ro root=/dev/hda1
```

Esse exemplo configura somente um único sistema operacional que o GRUB inicializará automaticamente (`default=0`) caso ele não receba qualquer entrada do teclado dentro de 10 segundos (`timeout=10`). O sistema de arquivos raiz para a configuração "Red Hat Linux" é o dispositivo GRUB default (hd0,0). O GRUB carregará o kernel em **/boot/vmlinuz-2.4.7-10** e exibirá uma tela splash no arquivo **/boot/grub/splash.xpm.gz** quando ele for carregado.

Fazendo boot múltiplo em PCs

Já que vários sistemas operacionais rodam em PCs, é relativamente comum configurar-se uma máquina capaz de inicializar através de vários sistemas diferentes. Para que isso funcione, precisamos configurar um boot loader para reconhecer todos os sistemas operacionais distintos presentes nos discos. Nas próximas seções, veremos alguns impedimentos comuns para sistemas de boot múltiplo e depois veremos alguns exemplos de configuração.

Cada partição de disco tem seu próprio boot loader de segundo estágio. Entretanto, há somente um MBR. Ao definirmos uma configuração boot múltiplo, temos que decidir qual boot loader irá se tornar o "mestre". Para melhor ou para pior, sua escolha normalmente será ditada pelos caprichos do sistema operacional em questão. LILO e GRUB são as melhores opções para um sitema que tenha uma partição Linux. As exceções a essa regra são o Windows NT e 2000 que, algumas vezes, possuem o seu próprio boot loader no MBR. Conhecemos sistemas nos quais o LILO e GRUB podem inicializar o Windows por meio do MBR; também sabemos de casos nos quais eles simplesmente não funcionam.

Note que o GRUB realmente não é superior ao LILO nesta stiuação em particular. Pode valer a pena instalar o GRUB na esperança de minimizar dores de cabeça relativas a ambientes de boot múltiplo.

Snow666

Pegadinhas de sistemas com boot múltiplo

Instalar um sistema com boot múltiplo pode fazer com que você arranque os seus cabelos. Esta seção destina-se a impedir que você fique careca.

Se estiver instalando um sistema com boot múltiplo que inclua uma versão do Windows (95, 98 ou Me) para o mercado consumidor, sempre instale o Windows antes de mais nada. As versões do Windows para o mercado consumidor são muito rudimentares e não percebem que pode haver outros sistemas operacionais já instalados numa mesma máquina. Estas versões do Windows sempre vão querer tomar a partição 1 do disco rígido e gravarão por cima de outros boot loaders durante a instalação.

A mesma regra se aplica ao Windows NT/2000: sempre instale o Windows primeiro. As razões para isso são ligeiramente distintas, porém o resultado final é o mesmo. O boot loader do NT/2000 realmente, mas realmente mesmo, quer ser instalado no MBR e ser o Boot Loader Onipotente para o sistema. Tentar resistir a isso é inútil.

Para que o boot loader do NT/2000 inicialize uma partição Linux, instale o Linux e depois inicialize-o usando um disquete ou CD-ROM. Você precisará extrair os primeiros 512 bytes da partição Linux (esta é a partição do registro de boot) e gravá-los num arquivo. Isso pode ser feito por meio do comando **dd**. Eis um exemplo:

```
# dd if=/dev/hda2 of=linux.bin bs=512 count=1
```

*A família de comandos **mtools** é útil para mover arquivos entre os ambientes Linux e Windows. Mais precisamente, os comandos **mtools** são utilizados para acessar sistemas de arquivos MS-DOS principalmente em disquetes, e, onde aplicável, emulando o respectivo comando do MS-DOS.*

A seguir você tem de copiar esse arquivo para a partição NT/2000 e acrescentar uma entrada à configuração do boot loader do NT que o informe como inicializar usando este arquivo. Não descrevemos o boot loader do NT neste livro, porém tudo o que você realmente precisa fazer é acrescentar uma linha no arquivo **C:\boot.ini** contendo o caminho do arquivo juntamente com um rótulo. Para o Linux, a linha seria algo como se segue:

```
C:\linux.bin="Linux"
```

Para maiores informações sobre o formato do arquivo **boot.ini**, dê uma olhada no Microsoft Knowledge Base em support.microsoft.com.

Se o Linux e o Windows NT/2000 estiverem coabitando e você estiver usando o LILO como seu boot loader, você precisará instalar o LILO na partição do Linux, pois o MBR já foi informado de tal. Aponte a linha `boot` no arquivo **lilo.conf** para a sua partição Linux para fazer essa alteração. Se, por exemplo, o Linux estiver na segunda partição do primeiro disco rígido IDE, essa linha ficará assim:

```
boot=/dev/hda2
```

Você precisará fazer essa modificação *antes* de copiar o boot loader de segundo estágio para um arquivo e o transferir para a partição NT. De fato, você precisará repetir todo o processo sempre que executar novamente o comando **lilo**.

Configuração de boot múltiplo com o LILO

Para configurar um sistema com boot múltiplo que use o LILO no MBR (por exemplo, Linux com Windows 98), comece com a configuração-padrão LILO conforme descrito anteriormente.. Depois você poderá retornar e adicionar entradas para os demais sistemas operacionais ao arquivo **/etc/lilo.conf**.

Eis a entrada de **lilo.conf** necessária para fazer o boot do Windows na primeira partição de seu primeiro disco IDE:

```
other = /dev/hda1
label = windows
table = /dev/hda
```

Um arquivo **lilo.conf** que faça o boot do Windows na partição 1, do Linux na partição 2 e do FreeBSD na partição 3 ficaria parecido com o seguinte:

```
boot=/dev/hda          # instala no MBR da primeira unidade IDE
root=/dev/hda1         # Especifica a partição-raiz
delay=20               # Aguarda 2 segundos para a escolha de inicialização
                       # por parte do usuário
image=/boot/vmlinuz-2.4.17
   root = /dev/hda2
   label=linux
   read-only
```

Snow666


```
other=/dev/hda1        # boot a partir da primeira partição
   label=windows
   table=/dev/hda
other=/dev/hda3        # boot a partir da terceira partição
   label=freebsd
   table=/dev/hda
```

Será necessário executar **lilo** novamente após adicionar todas essas entradas em **lilo.conf**. Lembre-se de executar **lilo -t** antes para testar o arquivo de configuração. Verifique a página 103 para obter maiores informações sobre partição.

Caso se sinta acuado, um plano de boot múltiplo alternativo é deixar o "outro" sistema operacional ter todos os espaços de bloco de boot "saudáveis" e fazer o boot da partição Linux em um disquete.

2.4 Inicializando em Modo Monousuário

Normalmente você entrará no modo monousuário do Linux através do LILO. Como atingir o prompt LILO varia de acordo com o hardware de seu sistema. Se você estiver usando hardware PC padrão, poderá chegar ao prompt LILO selecionando a opção de "linha de comando" na tela de boot gráfica nos sistemas Red Hat e SuSE. Em sistemas Debian, mantenha a tecla Shift pressionada imediatamente depois de o sistema ter finalizado seu próprio autoteste e verificações de memória.

No prompt LILO, digite o rótulo da configuração que quer fazer o boot (conforme especificado em **lilo.conf**) seguido de **-s** ou **single**. Por exemplo, a configuração-padrão que vem com o Red Hat é chamada "linux"; portanto, para inicializar esta configuração no modo monousuário, você deve usar:

```
LILO: linux single
```

O LILO aceita uma série de outras opções de linha de comando; a Tabela 2.1 ilustra alguns exemplos.

Tabela 2.1 Exemplos de opções no momento da inicialização do LILO.

Opção	Significado
root=/dev/foo	Informa o kernel para usar **/dev/foo** como dispositivo-raiz.
single	Inicializa no modo monousuário.
init=/sbin/init	Informa o kernel para usar **/sbin/init** como seu programa **init.**
ether=0,0,eth1	Faz com que o kernel sonde uma segunda placa Ethernet.
init=/bin/bash	Inicia somente o shell **bash**; útil para recuperação de emergência.

Como medida preventiva contra uma possível instabilidade do sistema[3], o diretório-raiz do sistema de arquivos começa a ser montado como somente para leitura. Isso pode ser contraproducente para sua tarefa caso esteja tentando solucionar um problema com um arquivo de configuração ou comando que resida no sistema de arquivos raiz ou se precisar executar um comando que modifique arquivos. Para solucionar esse problema, você terá de iniciar sua sessão monousuário remontando **/** no modo leitura/gravação. O comando **mount -o remount -w /** deve realizar isso.

2.5 Scripts de Inicialização

Após sair do modo monousuário (ou, na seqüência de boot automatizada, no ponto onde o shell monousuário teria sido executado), **init** executa os scripts de inicialização de sistema. Estes scripts são realmente apenas scripts shell corriqueiros que são interpretados por **sh** (bem, **bash**, na realidade). A localização, a organização e o conteúdo exatos dos scripts variam consideravelmente de sistema para sistema.

Algumas tarefas normalmente executadas nos scripts de inicialização são:

- Configurar o nome do computador
- Configurar a zona de horários
- Verificar os discos com **fsck** (somente no modo automático)
- Montar os discos do sistema

3. E contra usuários inescrupulosos que não deixam vestígios fingindo ser um administrador de sistemas.

www.snow666.blogspot.com
Snow666

- Eliminar arquivos antigos do diretório **/tmp**
- Configurar interfaces de rede
- Inicializar daemons e serviços de rede

A maioria dos scripts de inicialização são um tanto prolixos e geram uma descrição de tudo o que estão fazendo. Essa loquacidade pode ser de uma ajuda tremenda caso o sistema fique emperrado no meio do caminho do processo de inicialização ou se você estiver tentando localizar um erro em um dos scripts.

Em sistemas antigos, era uma prática comum modificar os scripts de inicialização para fazer que eles realizassem a coisa certa para um determinado ambiente. Nos dias de hoje, os scripts fornecidos com o sistema operacional devem ser genéricos o bastante para lidarem com praticamente qualquer configuração. Em vez de inserirmos no código dos scripts os detalhes de uma configuração local, os inserimos num arquivo de configuração distinto (ou conjunto de arquivos) que os scripts consultam. Os arquivos de configuração são normalmente simples mini-scripts **sh** que os scripts de inicialização incluem para definir os valores de certas variáveis de shell.

init e níveis de execução

Tradicionalmente, **init** define sete níveis de execução, cada um dos quais representando um determinado complemento de serviços que o sistema deve estar executando:

- O nível 0 é o nível no qual o sistema está completamente desligado.
- O nível 1 ou S representa o modo monousuário.
- Os níveis 2 a 5 são níveis multiusuário.
- O nível 6 é um nível de "reinicialização".

Os níveis 0 e 6 são especiais no sentido de que o sistema não pode realmente permanecer nestes níveis; o sistema é desligado ou reinicializado como efeito colateral da entrada nestes níveis. Na maioria dos sistemas, o nível de execução multiusuário normal é 2 ou 3. O nível de execução 5 é usado normalmente para os processos de login X Windows como **xdm**. O nível de execução 4 raramente é usado, e os níveis de execução 1 e S são definidos diferentemente em cada sistema.

O modo monousuário era tradicionalmente **init** nível 1. Ele desativava todos os processos de login remotos e multiusuário e certificava-se de que o sistema estava executando apenas o complemento de software mínimo. Porém, já que o modo monousuário fornece acesso root ao sistema, os administradores queriam que o sistema solicitasse a senha root sempre que ele fosse inicializado no modo monousuário. O nível S era criado para atender a esta necessidade: ele gera um processo que solicita a senha root. No Linux, o nível S atende somente a essa função e não é um fim por si só.

Parece haver mais níveis de execução definidos do que são estritamente necessários ou úteis. A explicação usual para isso é que uma mesa de telefonia tinha sete níveis de execução, de forma que se imaginou que um sistema UNIX devesse ter ao menos esta quantidade de níveis. O Linux na verdade suporta até dez níveis de execução, porém os níveis de 7 a 9 são indefinidos.

O arquivo **/etc/inittab** informa o **init** sobre o que fazer em cada um desses níveis. Seu formato varia de sistema para sistema, porém a idéia básica é de que **inittab** define comandos que devem ser executados (ou mantidos em execução) quando o sistema entra em cada nível.

Quando a máquina inicializa, **init** avança gradativamente do nível 0 para o nível de execução-padrão configurado em **/etc/inittab**. Para realizar a transição entre cada par de níveis de execução adjacentes, **init** executa as ações expressas para esta transição em **/etc/inittab**. A mesma progressão é feita em ordem reversa quando a máquina é desligada.

Infelizmente, a semântica do arquivo **inittab** é um tanto rudimentar. Para representar as facilidades do arquivo **inittab** em algo um pouco mais flexível, os sistemas Linux implementam uma camada de abstração adicional na forma de um script de "mudança de níveis de execução" (usualmente **/etc/init.d/rc**) que é chamado por meio de **inittab**. Por sua vez, esse script executa outros scripts em um diretório dependente de nível de execução para trazer o sistema para o seu novo estado.

Usualmente não é necessário para os administradores de sistema lidarem diretamente com **/etc/inittab**, pois a interface baseada em script é adequada para praticamente qualquer aplicativo. No restante deste capítulo ignoraremos tacitamente o arquivo **inittab** e o outro adesivo que adere **init** à execução de scripts de inicialização. Apenas tenha em mente que quando dizemos que **init** executa um script tal e tal, talvez a conexão não seja tão direta.

www.snow666.blogspot.com
Snow666

As cópias mestras dos scripts de inicialização residem no diretório **/etc/init.d**. Cada script é responsável por um daemon ou por um determinado aspecto do sistema. Os scripts reconhecem os argumentos **start** e **stop** para significar que o serviço com o qual eles lidam devem ser inicializados ou interrompidos. A maioria também reconhece **restart**, que é tipicamente o mesmo do que um **stop** seguido de um **start**. Na qualidade de administrador de sistemas, você pode iniciar e interromper serviços individuais executando manualmente os seus respectivos scripts **init.d**.

Eis, por exemplo, um script de inicialização simples que pode iniciar, interromper e reinicializar **sshd**:

```
#! /bin/sh
test -f /usr/local/sbin/sshd || exit 0
case "$1" in
      start)
         echo -n "Iniciando sshd: sshd"
         /usr/local/sbin/sshd
         echo "."
         ;;
      stop)
         echo -n "Interrompendo sshd: sshd"
         kill 'cat /var/run/sshd.pid'
         echo "."
         ;;
      restart)
         echo -n "Interrompendo sshd: sshd"
         kill 'cat /var/run/sshd.pid'
         echo "."
         echo -n "Iniciando sshd: sshd"
         /usr/local/sbin/sshd
         echo "."
         ;;
      *)
         echo "Uso: /etc/init.d/sshd start|stop|restart"
         exit 1
         ;;
esac
```

Embora os scripts em **/etc/init.d** possam iniciar e interromper serviços individuais, o script de controle mestre executado por **init** precisa de informações adicionais sobre quais scripts executar (e com quais argumentos) para entrar em qualquer nível de execução dado. Em vez de pesquisar diretamente no diretório **init.d** quando ele conduz o sistema a um novo nível de execução, o script mestre pesquisa num diretório chamado **rc***level***.d**, onde *level* é o nível de execução a ser inserido (por exemplo, **rc0.d**, **rc1.d** e assim por diante).

Os diretórios **rc***level***.d** contêm tipicamente links simbólicos que apontam de volta para os scripts contidos no diretório **init.d**. Os nomes desses links simbólicos começam com **S** ou **K** seguidos por um número e o nome do serviço que o script controla (por exemplo, **S34named**). Quando **init** faz a transição de um nível mais baixo para um mais alto, ele executa todos os scripts que iniciam com **S** em ordem numérica ascendente com o argumento **start**. Quando **init** faz a transição de um nível de execução mais alto para um mais baixo, ele executa todos os scripts que iniciam com **K** (de "kill") em ordem numérica descendente com o argumento **stop**. Dependendo do sistema, **init** pode pesquisar somente no diretório **rc***level***.d** apropriado para o novo nível de execução ou talvez ele também pesquise no diretório **rc***level***.d** para o nível de execução antigo para ver quais são as diferenças.

Para informarmos o sistema quando disparar um daemon, temos de colocar links simbólicos no diretório apropriado. Grande parte dos sistemas parece disparar a maioria de seus daemons de rede durante o nível de execução 2. Por exemplo, para informar o sistema para iniciar **sshd** durante o nível de execução 2 e para interromper o daemon de forma apropriada antes de desligar, o par de links a seguir seria suficiente:

```
# ln -s /etc/init.d/sshd /etc/rc2.d/S99sshd
# ln -s /etc/init.d/sshd /etc/rc0.d/K25sshd
```

A primeira linha informa o sistema para executar o script de inicialização **/etc/initi.d/sshd** como uma das últimas coisas a fazer ao entrar no nível de execução 2 e executar o script com o argumento **start**. A segunda linha informa o sistema para executar **/etc/init.d/sshd** relativamente tarde ao desligar o sistema e para executar o script com o argumento **stop**. Alguns sistemas tratam o desligamento e a reinicialização de maneira diferente, de modo que tere-

Snow666

mos de colocar um link simbólico no diretório **/etc/rc6.d**, bem como certificar-nos de que o daemon desliga apropriadamente quando o sistema é reinicializado.

Scripts de inicialização Red Hat

Os scripts de inicialização são uma das coisas que distinguem as diversas distribuições do Linux entre si. Os scripts do Red Hat são uma total confusão; eles contêm muitos comentários como

```
# stupid hack, but it should work (código estúpido, mas deve funcionar)
```

e

```
# this is broken! (isto está errado)
```

Em cada nível de execução, **init** chama o script **/etc/rc.d/rc** com o novo nível de execução como um argumento. **/etc/rc.d/rc** normalmente executa no modo "normal", no qual ele simplesmente faz o seu trabalho, porém ele também pode ser executado em modo de "confirmação" no qual ele nos indaga antes de executar cada script de inicialização individual. Use o comando **chkconfig** para gerenciar os links nos diretórios de nível de execução.

O Red Hat também possui um script **rc.local** muito parecido com aquele encontrado nos sistemas BSD. **rc.local** é o último script executado como parte do processo de inicialização. É melhor evitar suas próprias personalizações no arquivo **rc.local** pois este arquivo é sobrescrito pelos scripts de instalação.

Eis um exemplo de uma sessão de inicialização do Red Hat:

```
[kernel information]
INIT: version 2.77 booting
            Welcome to red Hat Linux
         Press 'I' to enter interactive startup

Mounting proc filesystem                                  [OK]
Setting clock (utc): Fri Feb 15 07:16:41 MST 2002         [OK]
Loading default keymap                                    [OK]
Activating swap partitions                                [OK]
...
```

Assim que vir a mensagem "Welcome to Red Hat Linux", poderá pressionar a tecla 'i' para entrar no modo de confirmação. Infelizmente o Red Hat não nos dá nenhuma confirmação que pressionamos a tecla correta. Ele prossegue alegremente para montar os sistema de arquivos locais, ativar as partições de swap, carregar os mapas de teclas e localizar os seus módulos de kernels. Somente após ele passar para o nível de execução 3, é que realmente começará a nos solicitar informações:

```
      Welcome to red Hat Linux
   Press 'I' to enter interactive startup
Mounting proc filesystem                                  [OK]
Setting clock (utc): Fri Feb 15 07:16:41 MST 2002         [OK]
Loading default keymap                                    [OK]
Activating swap partitions                                [OK]
Setting hostname redhat.synack.net                        [OK]
Checking root filesystem
/dev/hda1: clean, 73355/191616 files, 214536/383032 blocks [OK]
Remounting root filesystem in read-write mode             [OK]
Finding module dependencies                               [OK]
Checking filesystems                                      [OK]
Mounting local filesystems                                [OK]
Turning on user and group quotas for local filesystems    [OK]
Enabling swap space                                       [OK]
INIT: Entering running level: 3
Entering interactive startup
Start service kudzu (Y)es/(N)o/(C)ontinue? [Y]
```

A inicialização interativa e o modo monousuário começam no mesmo ponto no processo de inicialização. Quando o processo de inicialização é tão dividido que não podemos atingir este ponto de maneira segura, usamos um disquete de segurança para fazer a inicialização.

Snow666

Também podemos passar o argumento **init=/bin/sh** para o LILO para enganá-lo e fazê-lo executar um shell monousuário antes mesmo de **init** começar[4]. Se você adotar essa conduta terá de fazer toda a faxina de inicialização manualmente, inclusive dar o comando **fsck** manualmente e montar os sistema de arquivos locais.

Grande parte da configuração do processo de inicialização do Red Hat deve ser realizada através da manipulação dos arquivos de configuração em **/etc/sysconfig**. A Tabela 2.2 sintetiza a função dos itens no diretório **/etc/sysconfig**.

Vários dos itens contidos na Tabela 2.2 merecem comentários adicionais:

- O arquivo **hwconf** contém todas as informações de hardware. O obnóxio serviço Kudzu verifica para ver se adicionamos ou eliminamos qualquer dispositivo de hardware e nos pergunta o que fazer em relação às modificações. Provavelmente você desejará desabilitar esse serviço num sistema de produção, pois ele retarda o processo de inicialização sempre que detectar uma mudança na configuração de hardware, resultando em 30 segundos adicionais de downtime para cada mudança de hardware feita.

Tabela 2.2 Arquivos e subdiretórios do diretório /etc/sysconfig do Red Hat.

Arquivo/Diretório	Função do conteúdo
apmd	Relaciona argumentos para o daemon Advanced Power Management (gerenciamento avançado de energia).
clock	Especifica o tipo de relógio que o sistema possui (quase sempre UTC)[a]
console	Um diretório misterioso que está sempre vazio.
hwconf	Contém todas as informações de hardware do sistema. Usado pelo Kudzu.
i18n	Contém as configurações locais do sistema (formatos de data, linguagens etc.).
init	Configura a maneira pela qual as mensagens dos scripts de inicialização são exibidas.
keyboard	Configura o tipo de teclado (use "us" para o padrão de 101 teclas norte-americano).
mouse	Configura o tipo de mouse. Usado pelo X Windows e **gpm**.
network	Configura opções de rede globais (nome do host, gateway, encaminhamento etc.).
network-scripts	Um diretório que contém scripts acessórios e arquivos de configuração de rede.
pcmcia	Informa se deve ou não iniciar daemons PCMCIA e especifica opções.
sendmail	Configura opções para **sendmail**.

a. Se seu PC tiver vários boots, todas as apostas dependem de como o fuso horário do relógio deve ser configurado.

- O diretório **network-scripts** contém material adicional relacionado com a configuração de rede. As únicas coisas que devemos precisar modificar são os arquivos chamados **ifcfg**-*interface*. Por exemplo, **network-scripts/ifcfg-eth0** contém os parâmetros de configuração para a interface eth0. Ele configura o endereço IP da interface e as opções de rede. Veja as páginas 199 e 200 para obter maiores informações sobre configuração de interfaces de rede.
- O arquivo **sendmail** contém duas variáveis: `DAEMON` e `QUEUE`. Se a variável `DAEMON` estiver configurada em `yes`, o sistema dispara **sendmail** em modo daemon (**-bd**) quando o sistema é inicializado. `QUEUE` informa o **sendmail** quanto tempo ele deve esperar entre execuções de fila(**-q**); o padrão é uma hora.

Scripts de inicialização SuSE

Os scripts de inicialização são uma área na qual o SuSE realmente se destaca entre as variações do Linux. Os scripts de inicialização do SuSE são bem organizados, robustos e bem documentados. O pessoal que faz a manutenção desta parte do sistema operacional merece uma medalha de honra ao mérito.

Em cada nível de execução, **init** chama o script **/etc/init.d/rc** com o novo nível de execução na forma de um argumento. **/etc/rc.config** (que é mantido normalmente pela ferramenta de administração GUI do SuSE, o YaST) fornece os scripts de inicialização individuais com os dados de configuração na forma de variáveis de ambiente. Para subsistemas que requerem quantidades extraordinárias de configurações, arquivos individuais em **/etc/rc.config.d** fornecem os parâmetros necessários.

4. Certa vez tínhamos um arquivo de mapa de teclas corrompido e já que esse tipo de arquivo é carregado até mesmo no modo monousuário, o monousuário era inútil. Configurar **init=/bin/sh** era a única maneira de inicializar o sistema para um estado monousuário utilizável para corrigir o problema. Isso também pode ser um truque útil em outras situações.

Snow666

O arquivo **/etc/rc.config** é rico em opções que podemos personalizar para uma determinada situação em particular. Parâmetros básicos de sistema como o nome do host e interfaces de rede (com seus endereços IP associados) são definidos neste arquivo:

```
FQHOSTNAME="inura.toadranch.com"
IPADDR_0="192.168.1.101"
NETDEV_0="eth0"
IFCONFIG_0="192.168.1.101 broadcast 192.168.1.255 netmask 255.255.255.0"
```

Há também duas opções que especificam quais daemons disparar. Dicas úteis para a maioria das opções são fornecidas pelos comentários; por exemplo:

```
# inicia routed (para roteamento dinâmico – refira-se à página de manual routed) (yes/no)
# ATENÇÃO: iniciar routed provoca tráfego na rede a cada 30 segundos.
# Se seu host estiver conectado à Internet via linha discada, simplesmente não
# faz nenhum sentido ativar esta opção.
#
START_ROUTED="no"
```

Também há uma série de opções relacionadas com segurança no arquivo **/etc/rc.config** que podemos examinar para determinar se elas se aplicam ou não ao nosso ambiente. Veja o Capítulo 21, *Segurança*, para obter maiores informações sobre como estabelecer a segurança de seu sistema.

```
# Caso queira permitir logins root por meio de outras máquinas, configure
# ROOT_LOGIN_REMOTE em "yes".
#
ROOT_LOGIN_REMOTE="no"
```

Independentemente de optarmos por usar o YaST ou manter nossos scripts de inicialização manualmente, é uma boa idéia pesquisar através de **/etc/rc.config** e ponderar a respeito de seu conteúdo:

Eis como seria uma sessão típica de inicialização no SuSE (esta foi extraída de uma versão mais antiga):

```
[kernel information]
Inspecting /boot/System.map-2.2.18
Loaded 10080 symbols from /boot/System.map-2.2.18.
Symbols match kernel version 2.2.18.
Loaded 51 symbols from 1 module.
klogd 1.3-3, log source = ksyslog started.
<4>Linux version 2.2.18 (root@Pentium.suse.de) (gcc version 2.95.2 19991024
   (release)) #1 Wed Jan 24 12:28:55 GMT 2001
Checking file systems...
Parallelizing fsck version 1.19a (13-Jul-2000)
/dev/hda3: clean, 105679/2424832 files, 482130/4847613 blocks
/dev/hda1: clean, 31/6024 files, 4052/24066 blocks
Setting up the CMOS clock                                done
Mounting local file systems...
proc on /proc type proc (rw)
devpts on /dev/pts type devpts [rw,mode=0620,gid=5)
/dev/hda1 on /boot type ext2 (rw)                        done
Setting up timezone data                                 done
Setting up loopback device                               done
Setting up hostname                                      done
Configuring serial ports                                 done
Running /etc/init.d/boot.local                           done
```

Como pode ser observado, SuSE executa **/etc/init.d/boot.local** como o último script no processo de inicialização. Não recomendamos que você coloque esses comandos nesse arquivo.

Snow666

Scripts de inicialização Debian

Se o SuSE é o máximo em termos de um plano bem elaborado e executado para o gerenciamento de scripts de inicialização, o Debian é exatamente o oposto. Os scripts Debian são frágeis, não documentados e inacreditavelmente inconsistentes. Infelizmente, parece que a falta de uma maneira padronizada de configurar scripts resultou num caos nesse caso. Raios de Debian! Na verdade a política do Debian com relação aos scripts de inicialização é bem documentada. Toda a especificação dos scripts de inicialização do Debian pode ser encontrada em http://www.debian.org na seção Documentation. Traduções dessa política estão disponíveis em várias línguas, inclusive em português.

Em cada nível de execução, **init** chama o script **/etc/init.d/rc** com o novo nível de execução na forma de um argumento. Cada script é responsável por localizar as suas próprias informações de configuração que podem estar na forma de outros arquivos em **/etc**, em um subdiretório de **/etc** ou em algum outro ponto do script em si.

Se estiver procurando o nome de host do sistema, ele se encontra armazenado em **/etc/hostname**, que é lido pelo script **/etc/init.d/hostname.sh**. A interface de rede e os parâmetros de gateway-padrão estão armazenados em **/etc/network/interfaces**, que é lido pelo comando **ifup** chamado em **/etc/init.d/networking**. Algumas opções de rede também podem ser configuradas em **/etc/network/options**.

No Debian, os níveis de execução 0 (suspender) e 6 (reinicializar) são um pouco diferentes. Nesses dois níveis de execução, os links que iniciam com **S** ainda são chamados após aqueles que iniciam com **K**, mas eles também são chamados com um único argumento **stop**.

Também, se o nome de um script terminar com **.sh**, o script será incluido ("sourced") no nível de execução **S** em vez de ser executado em um subprocesso criado, mas ele será explicitamente executado pelo **sh** em todos os outros níveis de execução.

O Debian oferece ainda alguns utilitários para a manipulação dos scripts de inicialização. **update-rc.d** faz a manipulação dos links simbólicos nos diretórios **rc***level*.**d**. Por padrão, o **update-rc.d** iniciará serviços em cada um dos níveis de execução multiusuário (2, 3, 4 e 5) e irá pará-los no nível de execução de suspensão (0), no nível de execução monousuário (1) e no nível de execução de reinicialização (6). Para obter o comportamento-padrão na instalação de um novo script, o **update-rc.d** poderia ser chamado da seguinte maneira:

```
# update-rc.d pacote defaults
```

e para remover um script, o **update-rc.d** poderia ser chamado da seguinte maneira:

```
# update-rc.d pacote remove
```

Maiores informações sobre o **update-rc.d** são encontradas na sua página de manual.

Outro utilitário presente no Debian é o **start-stop-daemon** que pode ser utilizado para controlar a criação e o término de processos de nível de sistema. Usando-se as opções exec, -pidfile, -user e name, **start-stop-daemon** pode ser configurado para encontrar instâncias existentes de um processo em execução,

Utilizando o **start-stop-daemon**, um script de inicialização no Debian teria a seguinte forma:

```
#! /BIN/sh
DAEMON=/usr/sbin/daemon
NAME=daemon
DESC="algum daemon"

test -x $DAEMON || exit 0
set -e
case "$1" in
  start)
        echo -n "Starting $DESC: $NAME"
        start-stop-daemon -start -quiet -pidfile /var/run/$NAME.pid \
                -exec $DAEMON
        echo "."
        ;;
  stop)
        echo -n "Stopping $DESC: $NAME "
        start-stop-daemon -stop -quiet -pidfile /var/run/$NAME.pid \
                -exec $DAEMON
        echo "."
  restart|force-reload)
        echo -n "Restarting $DESC: $NAME"
```

Snow666


```
        start-stop-daemon -stop -quiet -pidfile \
                /var/run/$NAME.pid -exec $DAEMON
        sleep 1
        start-stop-daemon -start -quiet -pidfile \
                /var/run/$NAME.pid -exec $DAEMON
        echo "."
        ;;
  *)
        N=/etc/init.d/$NAME
        echo "Usage: $N {start|stop|restart|force-reload}" >&2
        exit 1
        ;;
esac
exit 0
```

O Debian fornece também um outro método de gerenciar os scripts de inicialização, em que toda a configuração é mantida em um único arquivo **/etc/runlevel.conf**. Basicamente, ele substitui aqueles links intermináveis e de difícil manutenção por um arquivo único: **/etc/runlevel.conf**. Para maiores informações sobre esse método, veja a página de manual e a documentação do pacote **file-rc.**

Boa sorte.

2.6 Reinicializando e Desligando

O buffer de sistemas de arquivos do Linux muda na memória e atualiza estas informações no disco somente esporadicamente. Esse esquema torna o I/O de disco mais rápido, porém também torna o sistema de arquivos mais suscetível a perda de dados quando o sistema é interrompido rudemente.

As máquinas UNIX e Linux tradicionais eram muito frágeis em relação a seu desligamento. Os sistemas modernos se tornaram menos sensíveis (especialmente quando usamos um sistema de arquivos robusto como o ext3fs), mas sempre é uma boa idéia desligar a máquina de maneira correta sempre que possível. O desligamento inadequado pode resultar desde problemas sutis e insidiosos a uma grande catástrofe.

Em sistemas operacionais voltados ao mercado de consumo, reinicializar o sistema operacional é um primeiro tratamento apropriado para quase todos os problemas. Num sistema Linux, é melhor pensar duas vezes antes de reiniciar o sistema. Os problemas no Linux tendem a ser mais sutis e mais complexos; portanto, reinicializar cegamente é eficiente apenas numa pequena porcentagem de casos. Os sistemas Linux também levam um longo tempo para inicializar e vários usuários poderão ser incomodados.

Talvez seja necessário reinicializarmos o sistema ao adicionarmos um dispositivo de hardware novo ou quando algum dispositivo de hardware existente se torna tão confuso que ele não é capaz de se reinicializar. Se modificarmos um arquivo de configuração que é usado somente em tempo de boot, temos de reinicializar para que nossas alterações entrem em efeito. Se o sistema estiver tão empacado a ponto de não conseguirmos fazer o login para realizarmos um diagnóstico apropriado do problema, obviamente não temos nenhuma alternativa a não ser reinicializar o sistema.

Algumas coisas podem ser alteradas e corrigidas sem a necessidade de reinicialização do sistema, porém sempre que modificamos um script de inicialização, devemos reinicializar simplesmente para garantir que o sistema retornará sem problemas.

Diferentemente do bootstrapping, que pode ser feito basicamente apenas de um modo, há uma série de maneiras para se desligar ou reinicializar o sistema. São elas:

- Desligar a força elétrica
- Utilizar o comando **shutdown**
- Utilizar os comandos **halt** e **reboot**
- Utilizar **telinit** para modificar o nível de execução de **init**
- Utilizar o comando **poweroff** para informar o sistema para desligar a força elétrica

Delisgar a força elétrica

Até mesmo num pequeno sistema Linux, desligar a força elétrica não é uma boa maneira de efetuar o processo de desligamento do sistema. Há o risco de perdermos dados e deixarmos os arquivos de sistema num estado inconsistente.

www.snow666.blogspot.com
Snow666

Algumas máquinas possuem uma "chave soft power" que significa que ao pressionarmos o botão Power, a máquina na verdade executa alguns comandos para executar uma seqüência de desligamento apropriada. Caso não esteja seguro se sua máquina fornece esse recurso, não pressione o botão Power para descobrir isso! É melhor você mesmo proceder à seqüência de desligamento.

Isso posto, não significa porém que desligar a força elétrica seja o fim do mundo. No caso de uma inundação ou incêndio, é aceitável desligar a força elétrica caso não tenha tempo suficiente para fazer toda a seqüência de desligamento. Salas de máquinas antigas normalmente tinham um botão de pânico que desligava tudo de uma só vez. Certo dia nossos administradores de sistema o dispararam sem querer com uma bolada. (Não me pergunte por que estavam jogando bola lá dentro!)

shutdown: a maneira gentil de desligar um sistema

shutdown é a maneira mais completa, segura e ponderada de iniciar o processo de parada ou de reinicialização ou de retornar ao modo monousuário.

Você poderá solicitar ao comando **shutdown** para aguardar um pouco antes de desligar o sistema. Durante o período de espera, **shutdown** envia mensagens (à la **wall**) para usuários logados em intervalos progressivamente menores, alertando-os do downtime. Por padrão, os alertas simplesmente dizem que o sistema está sendo desligado e dá o tempo restante em que ele ainda ficará no ar; também podemos fornecer uma pequena mensagem de nossa própria autoria. Essa mensagem poderia dizer por que o sistema está sendo desligado e estimar por quanto tempo ele ficará fora do ar (por exemplo, "o sistema estará de volta às 11:00"). Os usuários não poderão entrar no sistema quando um **shutdown** for iminente, porém eles verão essa mensagem caso ela tenha sido especificada.

shutdown permite especificar se a máquina deve parar (**-h**) ou reinicializar (**-r**) após o processo de desligamento estar completado. Também podemos especificar se queremos forçar a execução do comando **fsck** nos discos após uma reinicialização (**-F**) ou não (**-f**). Por padrão, o Linux pula as verificações **fsck** sempre que os sistemas de arquivos tiverem sido desmontados apropriadamente.

Por exemplo, um comando **shutdown** que lembre os usuários de uma manutenção programada e pare o sistema às 09:30 ficaria parecido com o seguinte:

```
shutdown -h 09:30 "Desativando o sistema para manutenção programada"
```

Também é possível especificarmos um tempo de desligamento relativo. Por exemplo, o comando a seguir realizaria o desligamento 15 minutos após sua execução:

```
shutdown -h +15 "Desativando o sistema para reparo de disco emergencial."
```

halt: uma maneira mais simples de desligar um sistema

O comando **halt** realiza as tarefas essenciais necessárias para desligar o sistema. Ele é chamado por **shutdown -h** mas também pode ser usado por si próprio. **halt** loga o desligamento, mata processos não essenciais, executa a chamada de sistema **sync** (chamado por e equivalente ao comando **sync**), aguarda as gravações de sistema de arquivos terminar e depois pára o kernel.

halt -n impede a chamada de **sync**. Ele é usado por **fsck** após reparar a partição-raiz. Se **fsck** não usar **-n**, o kernel poderia sobregravar os reparos feitos pelo **fsck** com versões antigas do superbloco que foram lançadas na memória (cache).

reboot: reinicialização rápida e suja

reboot é praticamente idêntico ao comando **halt**, porém ele faz com que a máquina seja reinicializada em vez de parar. **reboot** é chamado por **shutdown -r**. Como **halt**, ele suporta o flag **-n**.

telinit: modifica o nível de execução de init

Podemos usar **telinit** para direcionar **init** para um determinado nível de execução. Por exemplo,

```
# telinit 1
```

leva o sistema para o modo monousuário.

Ao usarmos **telinit**, não recebemos as mensagens de alerta amistosas ou o período "de lambuja" que temos no caso de **shutdown**; portanto, na maioria das vezes você vai querer evitá-lo. **telinit** é mais útil para testar mudanças feitas no arquivo **inittab**.

www.snow666.blogspot.com
Snow666

poweroff: solicita ao Linux para desligar o sistema

O comando **poweroff** é idêntico ao comando **halt**, exceto pelo fato de que após o Linux ter sido desativado, **poweroff** envia uma solitação ao sistema de gerenciamento de energia (em sistemas que possuem um) para desligar a alimentação principal do sistema. Este recurso facilita o desligamento de máquinas remotas (por exemplo, durante uma tempestade com raios).

Infelizmente, não há nenhum comando **poweron** correspondente. A razão para essa aparente omissão é deixada como exercício para o leitor.

2.7 EXERCÍCIOS

E2.1 Por que é importante executar o comando **lilo -t** antes de instalar o boot loader LILO? Como inicializamos um kernel cujo nome seja diferente de **vmlinuz**?

E2.2 Por que um sistema Linux não deve ser desligado com o botão Power do computador? Quais são alguns métodos alternativos?

E2.3 Se você tropeçar no cabo de alimentação e derrubar o sistema Linux, quais são as etapas necessárias para colocá-lo no ar novamente de forma satisfatória?

★ **E2.4** Explique o conceito de níveis de execução. Relacione os níveis de execução definidos no Linux e forneça uma breve descrição de cada um deles. Qual a relação entre o nível de execução 1 e o nível de execução S?

★ **E2.5** O que é bootstrapping? Relacione as etapas necessárias e discuta o que acontece em cada estágio.

★ **E2.6** Crie um script de inicialização para disparar o daemon "foo" (**/usr/local/sbin/foo**), um serviço de rede. Mostre como agregá-lo ao sistema para disparar automaticamente em tempo de boot.

★ **E2.7** Obtenha e instale o programa **mactime** de Dan Farmer e Wietse Venema (parte do kit de ferramentas TCT; refira-se à página 500). Execute **mactime** para criar um banco de dados inicial aos time stamps associados a seus sistemas de arquivos. Reinicialize a máquina. Rode **mactime** novamente e determine quais arquivos foram modificados por termos reinicializado a máquina. Que arquivos foram acessados, mas não modificados? (Requer acesso root.)

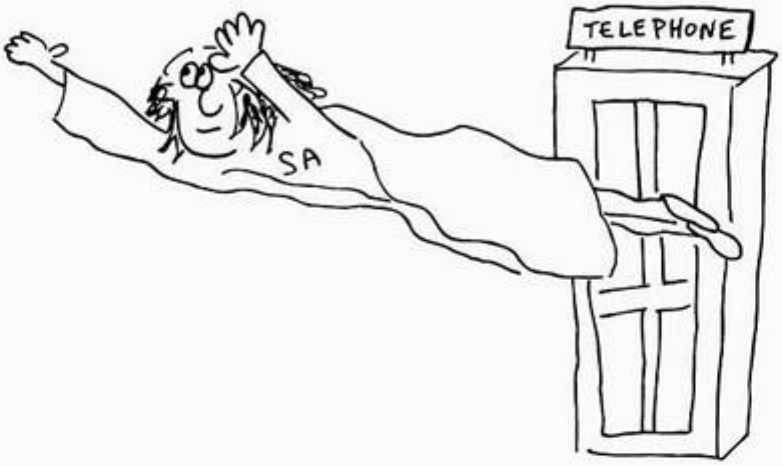
★★ **E2.8** Se um sistema se encontra no nível de execução 4 e você executar o comando **telinit 1**, que etapas serão tomadas por **init**? Qual será o resultado final do comando?

★★ **E2.9** Desenhe um gráfico de dependência que ilustre quais daemons têm de ser inicializados antes de outros daemons em seu sistema Linux.

★★ **E2.10** Relacione em ordem as etapas usadas para criar um sistema com vários sistemas operacionais que inclua o Linux e Windows XP (ou Windows 2000). Use GRUB e o boot loader do Windows.

Snow666

3 Poderes da Conta Root



Todo processo e arquivo num sistema Linux pertence a uma conta de usuário em particular. Os demais usuários não podem acessar esses objetos sem a permissão do proprietário, de modo que esta convenção ajuda a proteger os usuários contra as ações indevidas cometidas por outros, sejam elas intencionais ou acidentais.

Os arquivos e processos de sistema pertencem a um usuário fictício chamado "root" (ou *raiz*), também conhecido como superusuário. Como acontece com qualquer conta, a propriedade do root é protegida contra a interferência dos demais usuários. Para fazer alterações administrativas, você precisará usar um dos métodos de acesso à conta root descritos neste capítulo.

A conta root possui várias propriedades "mágicas". Ela pode atuar como o proprietário de qualquer arquivo ou processo. Também pode executar várias operações especiais que estão fora dos limites para os demais usuários. A conta é, ao mesmo tempo, muito poderosa, como também muito perigosa se cair em mãos inábeis ou maliciosas.

Este capítulo introduz aos administradores de sistemas os fundamentos do acesso superusuário. O Capítulo 21, *Segurança*, descreve como impedir acesso de superusuário indesejado e embaraçoso por parte de terceiros. O Capítulo 29, *Políticas e Diretrizes* cobre os aspectos políticos e administrativos relevantes.

3.1 Propriedade de Arquivos e Processos

Todo arquivo possui tanto um proprietário como um "proprietário de grupo". O proprietário do arquivo desfruta um privilégio especial que não é compartilhado com todo o mundo no sistema: a capacidade de modificar as permissões do arquivo. Em particular, o proprietário pode configurar as permissões num arquivo de maneira tão restritiva que ninguém mais poderá acessá-lo.[1] Tratamos do assunto permissões sobre arquivos no Capítulo 5, *O Sistema de Arquivos*.

Verifique a página 66 para obter maiores informações sobre grupos.

Embora o proprietário de um arquivo seja sempre uma única pessoa, muitas pessoas podem ser proprietárias de grupo do arquivo, desde que todas façam parte de um único grupo Linux. Os grupos são definidos no arquivo **/etc/group**.

O proprietário de um arquivo deve especificar o que os proprietários de grupo podem fazer ou não com ele. Esse esquema permite que arquivos possam ser compartilhados por membros do mesmo projeto. Por exemplo, usamos um grupo para controlar o acesso a arquivos-fonte para o site Web www.admin.com.

Ambas as propriedades de um arquivo podem ser determinadas através de **ls -l** *nome do arquivo*. Por exemplo:

1. Na realidade, as permissões podem ser configuradas de maneira tão restritiva que nem o próprio dono de um arquivo poderá acessá-lo; um recurso que é, na realidade, mais útil do que parece.

Snow666


```
% ls -l /staff/scott/todo
-rw---- 1 scott staff 1258 Jun 4 18:15 /staff/scott/todo
```

Esse arquivo é de propriedade do usuário "scott" e do grupo "staff".

O Linux, na verdade, controla os proprietários e grupos como números, e não como nomes na forma de texto. Os UIDs (User Identification Numbers, números de identificação de usuários) são associados a nomes de usuários no arquivo **/etc/passwd** e os GIDs (Group Identification Numbers, números de identificação de grupos) são associados a nomes de grupos no arquivo **/etc/group**.[2] Os nomes na forma de texto correspondentes aos UIDs e GIDs são definidos somente para conveniência de legibilidade de nossa parte. Quando comandos como **ls** querem exibir informações de propriedade num formato legível, eles têm de pesquisar cada nome no arquivo ou banco de dados apropriado.

Os processos não possuem duas, mas sim quatro identidades associadas a eles: um UID real e um efetivo e um GID real e um efetivo. Os números "reais" são usados para fins de contabilidade e os números "efetivos" são usados para determinação das permissões de acesso. Normalmente, os números reais e efetivos são os mesmos. O proprietário de um processo pode enviar os sinais de processo (veja a seção *Sinais*, no Capítulo 4) e também pode reduzir (degradar) a prioridade de escalonamento do processo.

Veja a seção Atributos de Arquivos no Capítulo 5 para obter maiores informações sobre bits de permissão.

Embora normalmente não seja possível para um processo alterar suas credenciais de propriedade, há uma situação especial na qual os IDs de grupos e usuários efetivos podem ser modificados. Quando um comando executado possui seu bit de permissões "setuid" ou "setgid" ativos, o UID ou GID efetivo do processo resultante podem ser configurados para o UID ou GID do arquivo contendo a imagem do programa, e não o UID e GID do usuário que executou o comando. Os privilégios do usuário são portanto "promovidos" somente para a execução desse comando específico.

O recurso setuid do Linux permite que programas sejam executados por usuários comuns fazendo uso da conta root de uma forma limitada e fortemente controlada. Por exemplo, o comando **passwd** que os usuários executam para modificar suas senhas de login é um programa setuid. Ele modifica o arquivo **/etc/shadow** (ou **/etc/passwd**) de uma maneira bem definida e depois finaliza. Obviamente, até mesmo essa tarefa limitada corre o risco de mau uso; desse modo **passwd** requer que os usuários provem que conhecem a senha da conta atual antes de ele concordar em fazer a modificação solicitada.

3.2 O Superusuário

A característica que define uma conta root é o seu UID igual a 0. O Linux não o impede de modificar o nome de usuário nesta conta ou de criar contas adicionais cujos UIDs sejam 0, porém ambas as ações são práticas terríveis. Tais mudanças apresentam uma tendência de criar inadvertidamente brechas na segurança do sistema. Elas também provocam confusão e rejeição quando outras pessoas têm de lidar com a maneira estranha que você configurou o sistema.

O Linux permite ao superusuário (isto é, qualquer processo cujo UID efetivo seja 0) executar qualquer operação válida em qualquer arquivo ou processo.[3] Além disso, algumas chamadas de sistema (solicitações ao kernel) podem ser executadas somente pelo superusuário. Alguns exemplos de tais operações restritas são:

- Modificar o diretório-raiz de um processo com **chroot**
- Criar arquivos de dispositivos
- Configurar o relógio do sistema
- Aumentar os limites de uso de recursos e as prioridades de processos
- Definir o nome de host (hostname) do sistema
- Configurar interfaces de rede
- Abrir portas de rede privilegiadas (aquelas com números abaixo de 1024)
- Desligar o sistema

Um exemplo dos poderes do superusuário é a capacidade de um processo de propriedade do root modificar o seu próprio UID e GID. O programa de **login** e seus equivalentes de sistema de janelas são um caso pertinente; o processo que nos solicita a senha ao fazermos a entrada num sistema é executado inicialmente como root. Se a senha

2. Alguns sistemas não armazenam mais essas informações sob a forma de arquivos de texto. Veja o Capítulo 18, *Compartilhando Arquivos de Sistema*, para obter maiores informações.
3. "Válida" é uma palavra evasiva importante nesse caso. Certas operações (como executar um arquivo no qual o bit de permissão de execução não está configurado) são proibidas até mesmo para o superusuário.

Snow666

e o nome de usuário introduzidos forem legítimos, o programa de login altera seu UID e GID para o nosso UID e GID e inicia nosso ambiente de usuário. Assim que um processo root tiver modificado seus direitos de propriedade para se tornarem um processo de usuário comum, ele não poderá recuperar seu estado privilegiado anterior.

3.3 Selecionando uma Senha Root

Veja o Capítuo 21 para obter maiores informações sobre quebra de senhas.

A senha root deve ter ao menos oito caracteres; senhas de sete caracteres são substancialmente mais fáceis de serem descobertas. Em sistemas que usam senhas DES, não adianta usar uma senha maior do que oito caracteres, pois somente os primeiros oito caracteres são significativos. Veja a seção *Senha Criptografada* no Capítulo 6 para obter maiores informações sobre como habilitar senhas MD5, que podem ser mais longas do que oito caracteres.

É importante que a senha root seja selecionada de maneira que não seja adivinhada com facilidade ou descoberta por tentativa e erro. Teoricamente, o tipo de senha mais seguro é formado por uma seqüência aleatória de letras, pontuação e dígitos. Mas pelo fato de esse tipo de senha ser difícil de ser memorizado e normalmente difícil de ser digitado, talvez não seja otimamente seguro se os administradores a anotarem em papel ou a digitarem de forma muito lenta.

Até recentemente, uma senha formada de duas palavras selecionadas aleatoriamente separadas por algum sinal de pontuação era um excelente compromisso entre segurança e facilidade de memorização. Infelizmente, tais senhas agora podem ser rompidas de modo relativamente rápido; hoje em dia não recomendamos o emprego deste esquema.

Nos dias de hoje, sugerimos que você forme uma senha root sintetizando uma frase "absurda chocante", definida por Grady Ward numa versão anterior da FAQ PGP Pass-phrase:

> *"Absurdo chocante" significa criar uma frase ou sentença curta que seja ao mesmo tempo absurda e chocante na cultura do usuário. Isto é, ela contém uma justaposição de idéias vulgarmente obscenas, racistas e impossíveis ou então extremistas. Esta técnica é permissível, pois a frase-senha, por sua natureza, jamais é revelada a alguém com sensibilidade de ser ofendido.*
>
> *Absurdo chocante tem poucas chances de ser duplicado, pois não descreve uma questão concreta que poderia ser acidentalmente redescoberta por uma outra pessoa. A evocação emocional faz com que fique difícil o criador se esquecer dela. Um exemplo de tal absurdo chocante poderia ser o seguinte: "Moluscos dão beijocas na minha genitália". O leitor, sem dúvida, será capaz de criar vários exemplos chocantes para uso próprio.*

Você pode reduzir uma frase desse tipo para uma senha registrando apenas a primeira letra de cada palavra ou através de algum processo de transformação semelhante. A segurança de senhas será aumentada enormemente se você incluir números, sinais de pontuação e letras maiúsculas.

Você deve trocar a senha root:

- Pelo menos a cada três meses
- Cada vez que alguém que conheça a senha saia da empresa
- Sempre que você julgar que a segurança pode ter sido comprometida
- Um dia em que você não esteja planejando divertir-se até altas horas da noite que terá esquecido a senha na manhã seguinte

3.4 Tornando-se o Root

Já que o root é simplesmente mais um usuário, poderíamos logar diretamente na conta root. Entretanto, isso acaba sendo uma má idéia. Só para começar, esse procedimento não deixa nenhum registro de quais operações foram efetuadas como root. Isso já é ruim o suficiente quando você se lembra que mexeu em algo que não devia às três da madrugada mas não consegue se lembrar o que foi que você alterou; é ainda pior quando um acesso não foi autorizado e você está tentando descobrir o que um intruso fez ao seu sistema. Uma outra desvantagem é que a situação de logar como root não deixa nenhum vestígio de quem estava realmente fazendo a operação. Se várias pessoas têm acesso à conta root, você não será capaz de informar quem a usou e quando a usou.

Por essas razões, a maioria dos sistemas permite que logins root sejam desabilitados em terminais e através da rede, ou seja, ele não pode ser feito em nenhum lugar exceto na console do sistema. Sugerimos que você use esses recursos. Veja *Terminais Seguros* no Capítulo 21 para descobrir que arquivo você precisa editar em seu sistema em particular.

Snow666

su: identidade de usuário substituta

Uma maneira ligeiramente melhor de acessar a conta root é usar o comando **su**. Se chamado sem qualquer argumento, **su** solicitará a senha root e depois iniciará um shell root. Os privilégios desse shell permanecem em efeito até que o shell seja encerrado (via <Control-D>) ou através do comando **exit**). **su** não registra nenhum comando executado como root, mas cria uma entrada de log atestando quem se tornou root e quando.

O comando **su** também pode substituir outras identidades que não sejam a do root. Algumas vezes, a única maneira de reproduzir ou depurar o problema de um usuário é usar **su** na conta dele para reproduzir a situação na qual ocorreu o problema.

Caso você conheça a senha de alguém, poderá acessar a conta da pessoa diretamente executando o comando **su** *nome_do_usuario*. Assim como **su** para root, lhe será solicitada a senha do usuário em questão. Também podemos usar **su** para entrar como root e depois **su** para uma outra conta; root pode acessar qualquer conta via **su** sem ter de fornecer uma senha.

É uma boa idéia adquirir o hábito de digitar o nome de caminho completo para o comando **su** (por exemplo, **/bin/su**) em vez de depender do shell para encontrar o comando para você. Isso lhe dará alguma proteção contra programas chamados **su** que podem ter sido sorrateiramente inseridos em seu caminho de busca com a intenção de capturar senhas.[4]

sudo: um su limitado

Já que os privilégios da conta superusuário não podem ser subdivididos, fica difícil dar a alguém a capacidade de realizar uma outra tarefa (backups, por exemplo) sem dar a essa pessoa livre acesso ao sistema. E se a conta root for usada por vários administradores, você realmente só terá uma vaga idéia de quem a está usando ou o que eles fizeram.

Nossa solução para esses tipos de problema é um programa chamado **sudo** que é mantido atualmente por Todd Miller. Ele é empacotado (vendido comercialmente) para Linux pela Red Hat, SuSE e Debian, mas também se encontra disponível na forma de código-fonte no site www.courtesan.com.

O programa **sudo** usa como argumento uma linha de comando a ser executada como root (ou como algum outro usuário restrito). **sudo** consulta o arquivo **/etc/sudoers**, que lista as pessoas que estão autorizadas a usar **sudo** e os comandos que elas têm permissão de executar em cada host. Se o comando proposto for permitido, **sudo** solicita a senha do *próprio usuário* e executa o comando.

Comandos **sudo** adicionais podem ser executados sem o "sudoer" ter de digitar uma senha até um período de cinco minutos (configurável) ter passado sem nenhuma atividade de **sudo**. Esse timeout serve como uma proteção modesta contra usuários com privilégios **sudo** que deixam terminais abandonados.

sudo mantém um log das linhas de comando que foram executadas, as pessoas que as solicitaram, o diretório no qual eles foram executados e os horários em que foram chamados. Essas informações podem ser registradas por syslog ou colocadas num arquivo de sua escolha. Recomendamos usar syslog para encaminhar as entradas de log para um host central "seguro".

Uma entrada de log para o usuário randy executando **sudo /bin/cat /etc/sudoers** ficaria parecida com o seguinte:

```
Dec 7 10:57:19 tigger sudo: randy: TTY=ttyp0 ;  PWD=/tigger/users/randy;
    USER=root ; COMMAND=/bin/cat /etc/sudoers
```

O arquivo **sudoers** é concebido para que uma única versão possa ser usada em vários hosts distintos ao mesmo tempo. Eis um exemplo típico:

```
# Define nomes alternativos para máquinas nos departamentos de Ciências
# da Computação & Física
Host_Alias   CS = tigger, anchor, piper, moet, sigi
Host_Alias   PHYSICS = eprince, pprince, icarus

# Define conjuntos de comandos
Cmd_Alias    DUMP = /sbin/dump, /sbin/restore
Cmd_Alias    PRINTING = /usr/sbin/lpc, /usr/bin/lprm
Cmd_Alias    SHELLS = /bin/sh, /bin/tcsh, /bin/csh, /bin/bash, /bin/ash,
   /bin/bsh
```

4. Pela mesma razão, recomendamos enfaticamente que você não inclua "." (o diretório atual) em seu caminho de busca do shell. Embora conveniente, essa configuração facilita a execução inadvertida de versões "especiais" dos comandos de sistema que um usuário ou invasor deixou lá como armadilha. Naturalmente, esse cuidado deve ser dobrado para o root.

Snow666


```
# Permissões
mark, ed     PHYSICS = ALL
herb         CS = /usr/local/bin/tcpdump : PHYSICS = (operator) DUMP
lynda        ALL = (ALL) ALL, !SHELLS
%wheel       ALL, !PHYSICS = NOPASSWD: PRINTING
```

As cinco primeiras linhas não comentadas definem os grupos de hosts e comandos que são referidos posteriormente nas especificações de permissão do arquivo. As listas poderiam ser incluídas literalmente nas especificações, porém o emprego de nomes alternativos faz com que os **sudoers** fiquem mais fáceis de serem lidos e compreendidos; isso também torna mais fácil suas atualizações no futuro. Também é possível definir nomes alternativos (aliases) para conjuntos de usuários e para conjuntos de usuários pelos quais os comandos possam vir a ser executados.

Cada linha de especificação inclui informações sobre:

- Os usuários aos quais a linha se aplica
- Os hosts nos quais a linha possa ser necessária
- Os comandos que os usuários especificados podem executar
- Os usuários pelos quais os comandos podem ser executados

A primeira linha de permissão se aplica aos usuários mark e ed nas máquinas do grupo `PHYSICS` (eprince, pprince e icarus). O nome alternativo de comando embutido `ALL` permite que eles executem qualquer comando. Já que nenhuma lista de usuário é especificada nos parênteses, **sudo** executará apenas comandos como root.

A segunda linha de permissão permite que o usuário herb execute **tcpdump** em máquinas CS e comandos de dump relacionados nas máquinas `PHYSICS`. Entretanto, os comandos de dump somente podem ser executados como operador e não como root. A verdadeira linha de comando que herb digitaria seria algo parecido com o seguinte:

```
% sudo -u operator /sbin/dump 0u /dev/hda2
```

A usuária lynda pode executar comandos como qualquer usuário em qualquer máquina, exceto que ela não é capaz de executar vários shells comuns. Isso significa que lynda realmente não pode efetivamente obter um shell root? Obviamente que não:

```
% cp -p /bin/csh /tmp/csh
% sudo /tmp/csh
```

Em termos genéricos, qualquer tentativa de permitir "todos os comandos exceto ..." é fadada ao insucesso, pelo menos em termos técnicos. Entretanto, ainda pode valer a pena configurar o arquivo **sudoers** dessa maneira como um lembrete que os shells root não são vistos com bons olhos. Isso pode desencorajar o emprego casual.

A linha final permite aos usuários do grupo wheel executarem os comandos **lpc** e **lprm** como root em todas as máquinas exceto eprince, pprince e icarus. Além disso, não é necessária nenhuma senha para executar os comandos.

Note que os comandos em **/etc/sudoers** são especificados com nomes de caminhos completos para evitar que terceiros executem seus próprios programas e scripts como root. Embora não seja apresentado nenhum exemplo acima, também é possível especificar os argumentos que são permitidos para cada comando. Na realidade, essa configuração simples é apenas uma pincelada da beleza e do esplendor que o arquivo **sudoers** oferece.

Para modificar **/etc/sudoers**, usamos o comando **visudo**, que faz uma verificação para garantir que ninguém mais esteja editando o arquivo, chama um editor e depois verifica a sintaxe do arquivo editado antes de instalá-lo. Essa última etapa é particularmente importante, pois um arquivo **sudoers** inválido poderia impedi-lo de aplicar um novo **sudo** para corrigi-lo.

O emprego de **sudo** apresenta as seguintes vantagens:

- O controle de contas é melhorado em muito devido ao logging dos comandos.
- Os operadores podem realizar tarefas sem a necessidade de privilégios root ilimitados.
- A verdadeira senha pode ser conhecida somente por uma ou duas pessoas.
- É mais rápido usar **sudo** do que executar **su** ou logar como root.
- Podem ser revogados privilégios sem a necessidade de modificar a senha root.
- É mantida uma lista canônica de todos os usuários com privilégios root.
- Há menos chances de um shell root ficar abandonado (sem interação).
- Um único arquivo pode ser usado para controlar o acesso a uma rede inteira.

Snow666

*Veja o Capítulo 21 para obter maiores informações sobre **crack**.*

Há uma série de desvantagens também. A pior delas é que qualquer brecha na segurança de uma conta pessoal de um usuário sudoer pode ser equivalente a uma brecha na conta root em si. Não há muito o que fazer para contra-atacar essa ameaça a não ser alertar os seus usuários sudoers para protegerem suas próprias contas como se fossem contas root. Você também pode executar **crack** regularmente nas senhas dos usuários sudoers para garantir que eles estejam optando por boas práticas na seleção de senhas.

O logging de comandos do **sudo** pode ser corrompido por truques como escapes de shell por meio de um programa autorizado ou através de **sudo csh** e **sudo su** se você os permitir.

3.5 Outros Pseudo-Usuários

Root é o único usuário que possui estado especial sob as vistas do kernel do Linux, porém vários outros pseudo-usuários são definidos pelo sistema. É comum substituir-se o campo de senha criptografado destes usuários especiais em **/etc/shadow** por um asterisco para que não se possa logar através de suas contas.

bin: proprietário legal dos comandos de sistema

Em alguns sistemas UNIX mais antigos, o usuário bin era dono dos diretórios que continham os comandos de sistema e a maioria dos comandos em si também. Esta conta é normalmente considerada supérflua nos dias atuais (ou talvez até mesmo ligeiramente insegura), de modo que sistemas modernos (inclusive o Linux) geralmente usam apenas a conta root. Por outro lado, agora que a conta bin é "padrão", também não podemos nos livrar dela.

daemon: proprietário de software de sistema não privilegiado

Arquivos e processos que fazem parte do sistema operacional, mas que não precisam ter como proprietários a conta root são algumas vezes dados para daemon. A teoria era que essa convenção ajudaria a evitar os danos em termos de segurança associados à propriedade por parte do root. Também existe um grupo chamado "daemon" por razões similares. Como a conta bin, a conta daemon não é muito usada pela maioria das distribuições do Linux.

nobody: o usuário genérico NFS

Verifique o Capítulo 17 para obter maiores informações sobre a conta nobody.

O NFS (Network File System, sistema de arquivos de rede) usa a conta nobody para representar usuários root em outros sistemas para fins de compartilhamento de arquivos. Para raízes remotas serem despojadas de seus poderes root, o UID 0 remoto tem de ser associado (mapeamento) a alguma coisa que não seja o UID 0 local. A conta nobody atua como o alterego genérico para essas raízes remotas.

Já que a conta nobody supostamente deve representar um usuário genérico e relativamente menos poderoso, ela não deve ter propriedade sobre quaisquer arquivos. Se nobody efetivamente possuir arquivos, raízes remotas serão capazes de assumir o controle sobre eles. Nobody não deve ser proprietário de nenhum arquivo!

Um UID -1 ou -2 era tradicional para nobody e esta convenção ainda é observada em algumas distribuições em que nobody possui UID 65534 (a versão complemento de dois de 16 bits de -2). Outras recebem um UID de numeração mais baixa como qualquer outro login de sistema, o que faz mais sentido.

3.6 Exercícios

★ **E3.1** Use o comando **find** com a opção **-perm** para localizar cinco arquivos setuid em seu sistema. Para cada arquivo, explique por que o mecanismo setuid é necessário para o comando funcionar apropriadamente.

★ **E3.2** Crie três frases-senha "absurdas e chocantes" mas guarde-as em segredo. Execute suas três frases-senha através de **md5sum** e informe os resultados. Por que é mais seguro compartilhar os resultados de MD5?

★ **E3.3** Enumere uma seqüência de comandos que modifiquem a entrada de senha de alguém e mostre como você poderia encobrir os seus rastros. Suponha que você tenha somente o poder **sudo** (todos os comandos permitidos, mas nenhum shell ou **su**).

★ **E3.4** Crie duas entradas para o arquivo de configuração **sudoers**:

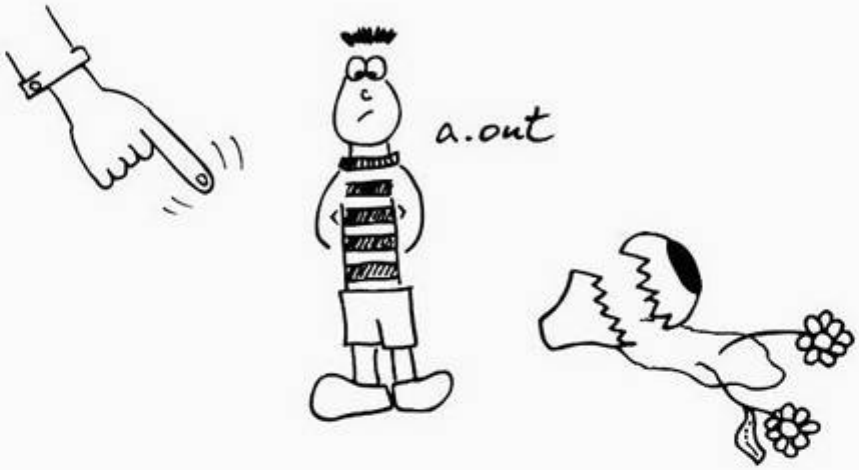
a) Uma entrada que permita aos usuários matt, adam e drew fazer a manutenção da impressora, desbloqueá-la e reinicializar daemons de impressão na máquina printserver.

b) Uma entrada que permita a drew, smithgr e jimlane matarem tarefas e reinicializarem as máquinas em student lab.

★ **E3.5** Instalar **sudo** configurado para enviar a você a fofoca dele sobre mau uso. Use-o para testar as entradas **sudo** da questão anterior com nomes de usuários e nomes de máquinas; verifique se **sudo** está registrando apropriadamente em syslog. Observe as entradas de syslog produzidas pelo seu teste. (Requer acesso root; muito provavelmente você terá também de ajustar **/etc/syslog.conf**.)

Snow666

4 Controlando Processos



Um processo é uma abstração usada pelo Linux para representar um programa em execução. É o objeto através do qual o uso de memória, tempo de processador e recursos de I/O de um programa podem ser gerenciados e monitorados.

É parte integrante da filosofia do Linux e do UNIX que a maior quantidade possível de trabalho possa ser feita dentro do contexto de processos, em vez de serem tratadas especialmente pelo kernel. Todos os processos de sistema e de usuário seguem as mesmas regras, de modo que você pode usar um único conjunto de ferramentas para controlar ambos.

4.1 Componentes de um Processo

Um processo é constituído de um espaço de endereços e um conjunto de estruturas de dados dentro do kernel. O espaço de endereços é um conjunto de páginas[1] de memória que o kernel marcou para ser empregado pelo processo. Ele contém o código e as bibliotecas que o processo está executando, as variáveis do processo, suas pilhas e várias informações adicionais que o kernel precisa enquanto um processo está em execução. Pelo fato de o Linux suportar memória virtual, não há necessariamente uma correlação entre a localização de uma página dentro de um espaço de endereços e sua localização dentro da memória física da máquina ou do espaço de swap.

As estruturas de dados internas do kernel registram vários tipos de informação sobre cada processo. Algumas das mais importantes são descritas a seguir:

- O mapa do espaço de endereços do processo
- O estado atual do processo (espera, parado, em execução etc.)
- A prioridade de execução do processo
- Informações sobre os recursos que o processo lançou mão
- A máscara de sinalização do processo (um registro de quais sinais são bloqueados)
- O proprietário do processo

Um processo também controla quais instruções a CPU está executando no momento em seu nome. O Linux permite que mais de um processo compartilhe um espaço de endereços atingindo, conseqüentemente, o efeito de

1. Páginas são as unidades nas quais a memória é gerenciada, normalmente 4 K, em PCs.

www.snow666.blogspot.com
Snow666

vários threads de execução. Até então, múltiplos threads de execução não têm tido muito impacto na administração de sistemas.

Muitos dos parâmetros associados a um processo afetam diretamente sua execução: a quantidade de tempo de processador que ele toma, os arquivos que ele pode acessar e assim por diante. Nas seções a seguir, discutimos o significado e a importância dos parâmetros mais interessantes sob a perspectiva de um administrador de sistemas. Esses atributos são comuns a todas as versões do UNIX e de Linux.

PID: número de identificação de processo

O kernel atribui um número de identificação exclusivo a cada processo. A maioria das chamadas de sistema e comandos que manipulam processos exige que especifiquemos um PID para identificar o objeto da operação. Os PIDs são atribuídos na ordem em que os processos são criados. Quando o kernel esgota os PIDs, ele reinicia em 1, pulando quaisquer PIDs que ainda se encontram em uso.

PPID: PID-pai

O Linux não fornece uma chamada de sistema que crie um processo executando num determinado programa. Em vez disso, um processo existente tem de se clonar para criar um processo novo. O clone pode então trocar o programa que ele está executando por um outro diferente.

Quando um processo é clonado, o processo original é chamado de pai e a cópia de filho. O atributo PPID de um processo é o PID do pai do qual ele foi clonado.[2]

Verifique a seção Número UID, no Capítulo 6, para obter maiores informações sobre UIDs.

UID e EUID: identificadores de usuário real e efetivo

O UID de um processo é o número de identificação do usuário da pessoa que o criou ou, mais precisamente, ele é uma cópia do valor EUID do processo-pai. Normalmente, somente o criador (também conhecido como "proprietário") e o superusuário têm permissão para manipular um processo.

O EUID é o ID de usuário "efetivo", um UID extra usado para determinar quais recursos e arquivos um processo tem permissão para acessar num dado momento. Para a maioria dos processos, o UID e o EUID são os mesmos, sendo a exceção usual programas aos quais são aplicados setuid.

Por que ter tanto UID quanto EUID? Simplesmente porque é útil manter uma distinção entre identidade e permissão e porque pode ser que um programa setuid não queira operar com permissões expandidas durante todo o tempo. O UID efetivo pode ser configurado e reinicializado para habilitar ou restringir as permissões adicionais que ele concede.

O Linux também controla um "UID salvo", que é uma cópia do EUID do processo no ponto onde ele começou a execução pela primeira vez. A menos que o processo tome medidas para eliminar esse UID salvo, ele permanecerá disponível para uso da mesma forma como o UID real ou efetivo. Um programa setuid escrito de maneira conservadora pode, entretanto, renunciar seus privilégios especiais para a maioria de sua execução, acessando-os somente em pontos específicos quando privilégios extras são necessários.

O Linux também define um parâmetro de processo FSUID que controla a determinação de permissões de sistema de arquivos. Ele não é usado freqüentemente.

As implicações desse sistema multi-UID pode ser bem sutil. Caso precise se aprofundar nos detalhes, duas fontes excelentes são o *Secure UNIX Programming FAQ* mantido por Thamer Al-Herbish em www.whitefang.com/sup e *Secure Programming for Linux and Unix HOWTO* de David A. Wheeler, disponível em www.linuxdoc.org (ambos em inglês)

GID e EGID: identificadores de grupo real e efetivo

Veja o Capítulo 6 para obter maiores detalhes sobre grupos.

GID é o número de identificação de grupo de um processo. O EGID está relacionado ao GID da mesma forma que o EUID está relacionado ao UID pelo fato de ele poder ser "atualizado" pela execução de um programa setgid. O Linux mantém um GID salvo similar na acepção ao UID salvo.

Em sistemas Linux, um processo pode ser um membro de vários grupos ao mesmo tempo. A lista de grupos é armazenada separadamente do GID e EGID distintos. As determinações de acesso normalmente levam em conta o EGID e a lista de grupo suplementar, mas não o GID.

2. Pelo menos inicialmente. Se o pai original expirar, **init** (process 1) se torna o novo pai. Veja a seção 4.2.

Snow666

Gentileza

A prioridade de escalonamento de um processo determina quanto tempo de CPU ele recebe. O kernel usa um algoritmo dinâmico para calcular prioridades, levando em conta a quantidade de tempo de CPU que um processo consumiu recentemente e o tempo que ele ficou aguardando para ser executado. O kernel também presta atenção a um valor normalmente chamado de "valor gentil" ou "gentileza", assim chamado pois ele informa quão gentil você está planejando ser com os demais usuários do sistema. Tratamos deste assunto com mais detalhes na seção 4.6.

Terminal de Controle

A maioria dos processos possui um terminal de controle associado a eles. O terminal de controle determina links-padrão para os canais de entrada-padrão, saída-padrão e erro-padrão. Ao iniciar um comando no shell, seu terminal normalmente se torna o terminal de controle do processo. O conceito de um terminal de controle também afeta a distribuição de sinais, que são discutidos na seção 4.3.

4.2 O Ciclo de Vida de um Processo

Para criar um processo, um processo faz uma cópia de si mesmo por meio da chamada de sistema **fork**. **fork** cria uma cópia do processo original que é em grande parte idêntica ao pai. O novo processo possui um PID distinto e possui suas próprias informações contábeis.

fork possui a propriedade exclusiva de retornar dois valores diferentes. Sob o ponto de vista do filho, ele retorna zero. Para o pai, por outro lado, é retornado o PID do filho recém-criado. Já que os dois processos são, sob outros aspectos, idênticos, ambos precisam examinar o valor retornado para descobrir que papel eles devem supostamente assumir.

Após um **fork**, o processo-filho normalmente usará uma das chamadas de sistema da família **exec** para começar a execução de um novo programa.[3] Essas chamadas mudam o texto do programa que o processo está executando e reinicializam os segmentos de dados e de pilha para um estado inicial predefinido. As várias formas de **exec** diferem somente na maneira pela qual elas especificam o ambiente e os argumentos de linha de comando a serem dados ao novo programa.

*Verifique o Capítulo 2 para obter maiores informações sobre inicialização e o daemon **init**.*

Quando o sistema é inicializado, o kernel cria e instala de maneira autônoma vários processos. O mais notável desses é **init**, que sempre é o processo número 1. **init** é responsável pela execução da maioria dos scripts de inicialização do sistema. Todos os processos, exceto aqueles que o kernel cria, são descendentes de **init**.

init também desempenha um importante papel no gerenciamento de processos. Quando um processo é completado, ele chama uma rotina chamada **_exit** para notificar o kernel de que ele está pronto para expirar. Ele fornece um código de saída (um inteiro) que informa o porquê de ele estar saindo. Por convenção, é usado 0 para indicar um término normal ou "bem-sucedido".

Antes de um processo ter a permissão para desaparecer completamente, o Linux requer que sua expiração seja reconhecida pelo processo-pai, o que o pai faz através de uma chamada a **wait**. O pai recebe uma cópia do código de saída do filho (ou uma indicação do porquê o filho foi extinto caso o filho não tenha abandonado voluntariamente) e também pode obter uma síntese do uso de recursos do filho caso ele queira.

Esse esquema funciona bem se os pais sobreviverem em relação aos seus filhos e se estiverem cientes sobre a necessidade de chamarem **wait** para que os processos extintos possam ser eliminados. Entretanto, se o pai for extinto primeiro, o kernel reconhecerá que nenhum **wait** virá no futuro e ajustará o processo para que o órfão se torne um filho de **init**. **init** aceita os processos-órfãos e executa o **wait** necessário para se livrar deles quando forem extintos.

4.3 Sinais

Sinais são solicitações de interrupção em nível de processo. São definidos cerca de 30 tipos distintos e eles são usados de várias maneiras:

- Podem ser enviados entre processos como um meio de comunicação.
- Podem ser enviados pelo driver de terminal para extinguir, interromper ou suspender processos quando teclas especiais como <Control-C> e <Control-Z> forem digitadas.[4]

3. Na verdade todas, exceto uma, são rotinas de biblioteca, e não chamadas de sistema.
4. As funções de <Control-Z> e <Control-C>podem ser reatribuídas a outras teclas através do comando **stty**, porém isso na prática é raro. Neste capítulo nos referimos a elas pelas suas associações convencionais.

www.snow666.blogspot.com
Snow666

- Podem ser enviados pelo administrador (via **kill**) para obter vários resultados.
- Podem ser enviados pelo kernel quando um processo comete uma infração como, por exemplo, uma divisão por zero.

Core dump é uma imagem da memória de um processo. Ele pode ser usado para depuração.

Quando um sinal é recebido, pode ocorrer uma de duas coisas. Se o processo receptor tiver designado uma rotina de manipulação para esse sinal em particular, o manipulador é chamado com informações sobre o contexto no qual o sinal foi entregue. Caso contrário, o kernel toma alguma atitude-padrão em nome do processo. A ação-padrão varia de sinal para sinal. Muitos sinais terminam o processo; alguns também geram um core dump.

Especificar uma rotina de manipulação para um sinal dentro de um programa é conhecido como "capturar" o sinal. Quando o manipulador encerra a sua ação, a execução é retomada do ponto no qual o sinal foi recebido.

Para impedir a chegada de sinais, os programas podem solicitar que eles sejam ignorados ou então bloqueados. Um sinal que é ignorado é simplesmente descartado e não possui nenhum efeito sobre o processo. Um sinal bloqueado é jogado em fila para remessa, porém o kernel não requer que o processo atue em seu nome até que o sinal tenha sido desbloqueado explicitamente. O manipulador para um sinal desbloqueado recentemente é chamado somente uma vez, mesmo que o sinal tenha sido recebido várias vezes enquanto a recepção estava bloqueada.

A Tabela 4.1 lista os sinais que todos os administradores devem conhecer. A convenção de letras maiúsculas para os nomes de sinais deriva da tradição da linguagem C. Às vezes você também poderá ver os nomes de sinais antecedidos por um prefixo SIG (por exemplo, SIGHUP) por razões similares.

Tabela 4.1 Sinais que todo administrador de sistemas deve conhecer.

#	Nome	Descrição	Padrão	Pode capturar?	Pode bloquear?	core dump?
1	HUP	Suspender	Encerrar	Sim	Sim	Não
2	INT	Interromper	Encerrar	Sim	Sim	Não
3	QUIT	Abandonar	Encerrar	Sim	Sim	Sim
9	KILL	Destruir	Encerrar	Não	Não	Não
[a]	BUS	Erro de barramento	Encerrar	Sim	Sim	Sim
11	SEGV	Falha de segmentação	Encerrar	Sim	Sim	Sim
15	TERM	Encerramento por parte do software	Encerrar	Sim	Sim	Não
[a]	STOP	Parar	Parar	Não	Não	Não
[a]	TSTP	Parada de teclado	Parar	Sim	Sim	Não
[a]	CONT	Continuar após parada	Ignorar	Sim	Não	Não
[a]	WHINCH	Modificado por janela	Ignorar	Sim	Sim	Não
[a]	USR1	Definido pelo usuário	Encerrar	Sim	Sim	Não
[a]	USR2	Definido pelo usuário	Encerrar	Sim	Sim	Não

a. Varia conforme a arquitetura de hardware; veja **man 7 signal**.

Há outros sinais que não são apresentados na Tabela 4.1, a maioria dos quais são usados para informar erros obscuros como "instrução ilegal". A manipulação-padrão para sinais como esses é a de encerrar com um core dump. A captura e o bloqueio são normalmente permitidos, pois alguns programas podem ser inteligentes o bastante para tentar limpar seja lá qual tenha sido o problema que provocou o erro antes de prosseguir.

Os sinais BUS e SEGV também são sinais de erro. Nós os incluímos na tabela, pois são muito comuns: 99% das vezes que um programa dá pau é, em última instância, um desses dois sinais que finalmente o derruba. Por si só, os sinais não têm nenhum valor específico em termos de diagnóstico. Ambos indicam uma tentativa de usar ou acessar a memória de forma inadequada.

A maioria dos emuladores de terminais enviará um sinal WINCH quando houver mudanças em seus parâmetros de configuração (como, por exemplo, o número de linhas no terminal virtual). Essa convenção permite a programas que entendem emuladores (editores de texto, em sua maioria) se autoreconfigurarem em resposta a mudanças. Se você não obtiver um redimensionamento de janelas apropriado, certifique-se de que WINCH esteja sendo gerado e propagado corretamente.

www.snow666.blogspot.com
Snow666

Os sinais chamados KILL e STOP não podem ser capturados, bloqueados ou ignorados. O sinal KILL destrói o processo receptor e STOP suspende sua execução até que um sinal CONT seja recebido. CONT pode ser capturado ou ignorado, mas não bloqueado.

TSTP é uma versão "soft" de STOP que poderia ser melhor descrita como uma solicitação de parada. É o sinal gerado pelo driver do terminal quando digitamos um <Control-Z>. Programas que capturam esse sinal normalmente limpam seus estados, depois enviam a eles próprios um sinal STOP para completar a operação de parada. Alternativamente, TSTP poderia simplesmente ser ignorado, para impedir que o programa seja interrompido pelo teclado.

Os sinais KILL, INT, TERM, HUP e QUIT podem parecer todos iguais, porém suas aplicações são, na verdade, bem diferentes. É lamentável que terminologia tão vaga tenha sido escolhida para eles. Eis um guia de decodificação:

- KILL não pode ser bloqueado e encerra um processo em nível de sistema operacional. Um processo jamais "recebe" este sinal.
- INT é o sinal enviado pelo driver de terminal quando digitamos <Control-C>. É uma solicitação para encerrar a operação atual. Programas simples poderiam sair (se eles capturarem o sinal) ou simplesmente permitir que eles próprios sejam eliminados, que é o padrão caso o sinal não seja capturado. Programas que possuem uma linha de comando ou modo de entrada devem interromper o que estavam fazendo, fazer uma limpeza e aguardar novamente por uma entrada de usuário.
- TERM é uma solicitação para terminar completamente a execução. Espera-se que o processo receptor limpe o seu estado e saia.
- HUP possui duas interpretações comuns. A primeira delas é como uma solicitação de reinicialização por vários daemons. Se um daemon for capaz de reler seu arquivo de configuração e ajustar-se às alterações sem a necessidade de reinicialização, geralmente pode ser usado um HUP para disparar esse tipo de comportamento. Em segundo lugar, os sinais HUP são algumas vezes gerados pelo driver de terminal numa tentativa de "limpar" (isto é, extinguir) os processos agregados a um determinado terminal. Isso pode acontecer, por exemplo, quando uma sessão de terminal é concluída ou quando uma conexão de modem cai de repente (daí o termo "ficar pendurado" ou "suspenso"). Os detalhes variam de sistema para sistema.
- Os shells na família de shells C (**tcsh** et al.) normalmente tornam os processos em segundo plano imunes a sinais HUP para que eles possam continuar a serem executados após o usuário sair do sistema. Os usuários de shells tipo Bourne (**ksh**, **bash** etc.) podem emular esse comportamento através do comando **nohup**.
- QUIT é similar à TERM exceto pelo fato de seu padrão ser gerar um core dump caso não seja capturado. Alguns poucos programas canibalizam este sinal e o interpretam como tendo outro significado.

Os sinais USR1 e USR2 não possuem um significado fixo. Eles se encontram disponíveis para os programas usarem da forma que bem entenderem. Por exemplo, o servidor Web Apache interpreta o sinal USR1 como sendo uma solicitação para reinicializar graciosamente.

4.4 Kill e Killall: Enviando Sinais

Como o próprio indica, o comando kill é usado mais freqüentemente para encerrar um processo. kill é capaz de enviar qualquer sinal, porém, como padrão, ele envia um TERM. **kill** pode ser usado por usuários comuns em seus próprios processos ou pelo superusuário em qualquer processo. A sintaxe é

```
kill [-signal] pid
```

onde *signal* é o número ou nome simbólico do sinal a ser enviado (conforme indicado na Tabela 4.1) e *pid* é o número de identificação do processo de destino. Um *pid* -1 transmite o sinal a todos os processos, exceto **init**.

Um **kill** sem qualquer número de sinal não garante que o processo será extinto, pois o sinal TERM pode ser capturado, bloqueado ou ignorado. O comando

```
kill -9 pid
```

"garantirá" que o processo se extinguirá, pois o sinal 9, KILL, não pode ser capturado. Colocamos a palavra garantirá entre aspas, pois os processos algumas vezes podem se tornar tão empacados que nem mesmo KILL os afetará (normalmente devido a alguma degeneração de I/O como, por exemplo, a espera por um disco que parou de girar). A reinicialização normalmente é a única saída para se livrar de tais processos indisciplinados.

Snow666

A maioria dos shells tem sua própria implementação de **kill** embutida que obedece a sintaxe descrita. De acordo com a página de manual para o comando **kill** independente, o número ou nome de sinal deve ser precedido, na verdade, por um flag **-s** (por exemplo, **kill -s HUP** *pid*). Mas já que alguns shells não entendem essa versão de sintaxe, sugerimos que você fique com a forma **-HUP**, que também o comando **kill** independente compreende. Dessa maneira, você não precisará se preocupar com a versão de **kill** que está utilizando.

Se não conhecer o PID do processo que deseja sinalizar, normalmente o pesquisamos através do comando **ps**, que é descrito a partir da página 45. Uma outra opção é usar o comando **killall**, que executa essa pesquisa para nós. Por exemplo, para fazermos com que o daemon **xinetd** restaure sua configuração, deveríamos executar o seguinte:

```
% sudo killall -USR1 xinetd
```

O comando **kill** corriqueiro na verdade tem um recurso similar, porém ele não parece ser tão inteligente quanto **killall** no critério de busca de nomes de comandos. Fique com **killall**.

4.5 Estados de Processos

Um processo não é automaticamente elegível para receber tempo de CPU simplesmente porque ele existe. Há, basicamente, quatro estados de execução sobre os quais você deve estar ciente e que são listados na Tabela 4.2.

Tabela 4.2 Estados de processos.

Estado	Significado
Executável	O processo pode ser executado.
Dormente	O processo está aguardando algum recurso.
Zumbi	O processo está tentando se destruir.
Parado	O processo é suspenso (não há permissão para ser executado).

Um processo executável é aquele que está pronto para ser executado toda vez que houver tempo de CPU disponível. Ele obteve todos os recursos necessários e está apenas esperando tempo de CPU para processar seus dados. Assim que o processo fizer uma chamada de sistema que não pode ser completada imediatamente (como uma solicitação de leitura de parte de um arquivo), o Linux o colocará no estado de dormência.

Os processos em estado de dormência estão aguardando a ocorrência de um evento específico. Daemons de sistema e shells interativos passam grande parte de seu tempo no estado de dormência, aguardando entrada via terminal ou conexões de rede. Já que um processo em estado de dormência é efetivamente bloqueado até que sua solicitação seja atendida, ele não obterá nenhum tempo de CPU a menos que receba um sinal. Zumbis são processos que terminaram a execução, porém ainda não tiveram seus estados coletados.

Processos parados são administrativamente proibidos de serem executados. Os processos são interrompidos no recebimento de um sinal STOP ou TSTP e são reiniciados com CONT. Ser interrompido é similar a adormecer, porém não há nenhum jeito de sair do estado interrompido a não ser ter algum outro processo que nos faça acordar (ou nos mate).

4.6 Nice e Renice: Influência sobre a Prioridade de Programação

A "gentileza" de um processo é uma dica numérica para o kernel em relação a como o processo deve ser tratado no que diz respeito a outros processos lutando por recursos de CPU. O nome estranho é derivado do fato de que ele determina quão gentil seremos com os demais usuários do sistema. Um valor nice (gentil) alto significa baixa prioridade para o seu processo: você será gentil. Um valor nice baixo ou negativo significa alta prioridade: você não será gentil. O intervalo de valores nice permitidos vai de -20 a +19.

A menos que o usuário tome alguma ação em especial, um processo recém-criado herda o valor nice de seu processo-pai. O proprietário do processo pode aumentar seu valor nice, mas não diminuí-lo. Essa restrição impede que processos com baixa prioridade sejam gerados de filhos de alta prioridade. O superusuário possui liberdade total na configuração de valores nice e pode até mesmo configurar a gentileza de um processo com um valor tão baixo que impedirá que outros processos possam ser executados.

Veja o Capítulo 26 para obter maiores informações sobre desempenho.

Configurar manualmente prioridades de processos está se tornando rapidamente uma coisa do passado. Quando o UNIX rodava nos frágeis sistemas das décadas de 70 e 80, o desempenho era em grande parte afetado significativamente pelo processo que se encontrava na CPU. Hoje em dia, com poder de CPU mais do que adequado na maioria

Snow666

dos desktops, o escalonador normalmente faz um bom trabalho no atendimento aos processos. Entretanto, o desempenho de I/O não acompanhou a velocidade cada vez maior das CPUs, e o maior gargalo na maioria dos sistemas passou a ser as unidades de disco. Infelizmente, o valor nice de um processo não tem nenhum efeito sobre o gerenciamento do kernel de sua memória ou I/O; processos com valores nice altos ainda podem monopolizar uma fatia desproporcional desses recursos.

O valor nice de um processo pode ser configurado no momento da criação pelo comando **nice** e pode ser ajustado durante a execução com o comando **renice**. **nice** usa uma linha de comando como argumento e **renice** usa um PID ou um nome de usuário. De maneira um tanto confusa, **renice** requer uma prioridade absoluta, porém **nice** quer um incremento de prioridade que ele então acrescenta ou subtrai da prioridade atual do shell.

Alguns exemplos:

```
% nice -n 5 ~/bin/longtask          // Diminui a prioridade (aumenta nice) de 5
% renice -5 8829                    // Configura o valor de nice em -5
% sudo renice 5 -u boggs            // Configura o valor nice dos processos
                                    // de boggs em 5
```

Para complicar as coisas ainda mais, uma versão do **nice** é embutida no shell C e alguns outros shells comuns (mas não **bash**). Se não digitarmos o caminho completo para o comando nice, obteremos a versão do shell em vez da do sistema operacional. Isso pode ser confuso, pois o **nice** do shell e o comando **nice** independente usam sintaxes diferentes: o shell quer seu incremento de prioridade expresso na forma *+incr* ou *-incr*, porém o comando dedicado quer um flag **-n** seguido por um incremento de prioridade.[5]

O processo ao qual o **nice** é mais comumente aplicado no mundo moderno é **xntpd**, o daemon de sincronização de relógio. Já que a prontidão da CPU é crítica para esse caso, ele normalmente é executado com um valor nice cerca de 12 unidades abaixo do padrão (isto é, numa prioridade mais alta do que o normal).

Se um processo ficar frenético e levar a média de carga do sistema para 65, talvez seja necessário usar **nice** para iniciar um shell de alta prioridade antes de executar comandos para investigar o problema. Caso contrário, esses comandos talvez jamais tenham uma chance de serem executados.

4.7 ps: Monitorar Processos

ps é a principal ferramenta do administrador de sistemas para monitoramento de processos. Podemos usá-lo para exibir o PID, o UID, a prioridade e o terminal de controle dos processos. Ele também dá informações sobre quanta memória foi consumida e seu estado atual (em execução, interrompido, dormente etc.). Os zumbis aparecem numa listagem do **ps**s como <defunct>.

O comportamento de **ps** tende a variar enormemente entre as variantes do UNIX, e muitas implementações se tornaram bem complexas ao longo dos últimos anos. Num esforço de acomodar as pessoas que estão acostumadas com comandos **ps** de outros sistemas, o Linux fornece uma versão trisexual e hermafrodita que entende muitos outros conjuntos de opções de implementações e usa uma variável de ambiente para informar a ele que personalidade assumir.

Não fique assustado com toda essa complexidade: ela está lá principalmente para os desenvolvedores de kernels, não para administradores de sistemas. Embora usaremos **ps** freqüentemente, apenas precisaremos saber alguns truques específicos.

Podemos obter uma visão geral dos processos em execução no sistema via **ps aux**. Eis um exemplo (eliminamos a coluna START para que o exemplo coubesse na página e selecionamos somente uma amostra das linhas de saída):

```
% ps aux
  USER   PID   %CPU  %MEM  VSZ   RSS   TTY       STAT  TIME  COMMAND
  root     1   0.0   0.0   1344  76    question  S     0:05  init [3]
  root     2   0.0   0.0      0  0     question  SW    0:02  [kflushd]
  root     3   0.0   0.0      0  0     question  SW    0:01  [kupdate]
  root   477   0.0   0.4   2284  552   question  S     0:01  xinetd-reuse-pi
  root   567   0.0   0.3   3240  408   question  S     0:02  Sendmail:accepti
  root   770   0.0   0.0   1572  120   question  S     0:00  Crond
  root   966   0.0   0.1   2512  208   question  S     0:00  /usr/sbin/sshd
  root   307   0.0   1.5   1944  1936  question  SL    0:00  ntpd
```

5. Na verdade, é até pior do que isso: o **nice** dedicado interpretará **nice -5** como um incremento positivo de 5, ao passo que o **nice** embutido interpretará essa mesma forma como um incremento negativo de 5.

Snow666


```
 trent  1330  0.0   1.0   2464  1352  pts/5      S    0:00  -csh
 named  1886  0.0   1.3   3232  1732  question   S    0:04  named -unamed
 boggs  2148  0.0   0.5   1616  764   pts/3      T    0:00  man iptables
 boggs  2150  0.0   0.7   2060  920   pts/3      T    0:00  sh -c/bin/gunzip
 boggs  2151  0.0   0.3   1680  476   pts/3      T    0:00  /bin/gunzip -c/v
 boggs  2152  0.0   0.6   1868  820   pts/3      T    0:00  /usr/bin/less -is
 garth  5279  0.0   1.1   2616  1472  pts/7      S    0:00  -csh
apache  5392  0.0   4.1   8248  5356  question   S    0:00  /usr/sbin/httpd -
apache  5400  0.0   4.1   8248  5356  question   S    0:00  /usr/sbin/httpd -
...
```

O significado de cada campo é explicado na Tabela 4.3.

Como pode ser observado neste exemplo, **ps** recorta os comandos cuidadosamente para que cada processo ocupe uma linha. Para impedir o recorte e ver o máximo de informações possíveis que ps é capaz de fornecer, inclua a opção **w** (de "wide") na linha de comando.

Um outro conjunto útil de argumentos é **lax**, que fornece mais informações técnicas. Ele também é mais rápido na execução, pois não tem de traduzir cada UID num nome de usuário — a eficiência pode ser importante se o sistema já estiver atolado por algum outro processo. **ps** geralmente é bastante custoso para ser executado.

Veja a seguir um breve exemplo de como **ps lax** inclui campos como o PPID (Parent Process ID), valor nice (NI) e nome ou função do kernel na qual o processo está dormente (WCHAN).

Tabela 4.3 Explicação da saída gerada por **ps aux**.

Campo	Conteúdo
USER	Nome do usuário do proprietário do processo
PID	ID (identificador) do processo
%CPU	Porcentagem dos recursos de CPU que este processo está usando
%MEM	Porcentagem da memória real que este processo está usando
VSZ	Tamanho virtual do processo
RSS	Resident Set Size (número de páginas na memória)
TTY	ID (identificador) de terminal de controle
STAT	Estado de processo atual R = Executável D = Espera no disco (ou a curto prazo) S = Dormente (<20 seg) T = Rastreado ou interrompido Z = Zumbi Flags adicionais: W = Processo paginado em disco < = O processo tem uma prioridade maior do que a normal N = O processo tem uma prioridade menor do que a normal L = Algumas páginas são bloqueadas no núcleo (kernel)
START	Horário em que o processo foi iniciado
TIME	Tempo de CPU que o processo consumiu
COMMAND	Nome do comando e argumentos[a]

a. Os argumentos podem ser truncados; acrescente o argumento **ww** para evitar isto. Os programas são capazes de modificar esta informação, de modo que não é necessariamente uma representação acurada da verdadeira linha de comando.

```
% ps lax
  F   UID   PID  PPID  PRI   NI   VSZ  RSS  WCHAN   STAT  TIME  COMMAND
100     0     1     0    0    0  1344   76  do_sel  S     0:05  init[3]
040     0     2     1    0    0     0    0  bdfus   SW    0:02  [kflushd]
040     0     3     1    0    0     0    0  kupdat  SW    0:01  [kupdate]
140     0   477     1    0    0  2284  552  do_sel  S     0:01  xinetd-reu
100  1011  8239  8238    0    0  2608 1496  read_c  S     0:00  -csh
140    25 11886     1    0    0  3232 1732  do_sel  S     0:04  named-una
000  1011 12148  8239    0    0  1616  764  do_sig  T     0:00  maniptable
000  1011 12150 12148    0    0  2060  920  do_sig  T     0:00  sh-c/bin/
```

Snow666


```
000  1011  12151   12150   0    0    1680   476  do_sig   T     0:00  /bin/gunzip
000  1011  12152   12150   0    0    1868   820  do_sig   T     0:00  /usr/bin/le
...
```

4.8 top: Monitoramento Ainda Melhor dos Processos

Já que comandos como **ps** oferecem apenas um instantâneo de seu sistema num dado momento, normalmente fica difícil ter uma compreensão do todo. O comando **top** fornece um sumário atualizado regularmente dos processos ativos e o emprego de seus recursos.

Por exemplo:

```
5:45pm up 30 days, 22:25, 9 users, load average: 0.00, 0.00, 0.00
74 processes: 69 sleeping, 1 running, 0 zombie, 4 stopped
CPU states: 0.0% user, 0.0% system, 0.0% nice, 0.3% idle
Mem: 127884K av, 122676K used, 5208K free, 73344K shrd, 68260K buff
Swap: 265032K av,  5056K used, 259976K free             21524K cached

 PID  USER     PRI   NI    SIZE   RSS  SHARE  STAT  %CPU  %MEM  COMMAND
6785  garth     16    4    1056  1056  832     R N  0.9   0.8   top
   1  root       0    0     120    76  56        S  0.0   0.0   init
   2  root       0    0       0     0  0        SW  0.0   0.0   kflushd
 337  root       0    0     496   180  152       S  0.0   0.1   klog
 352  rpc        0    0     508   500  416       S  0.0   0.3   portmap
 379  rpcuser    0    0     112     0  0        SW  0.0   0.0   rpc.stad
 431  daemon     0    0     116    56  40        S  0.0   0.0   atd
 477  root       0    0     636   564  392       S  0.0   0.4   xinetd
 507  lp         0    0     128     0  0        SW  0.0   0.0   lpd
 567  root       0    0     708   408  304       S  0.0   0.3   sendmail
 707  root       0    0    5740  5140  4976      S  0.0   4.0   http
...
```

Como padrão, a tela é atualizada a cada 10 segundos. Os processos mais ativos aparecem no alto. **top** também aceita entrada pelo teclado e permite que enviemos sinais e alteremos os valores de nice dos processos, para que possamos observar como nossas ações afetam a condição geral da máquina.

top tem de consumir uma pequena porção dos recursos de CPU para mostrar uma atualização a cada 10 segundos. Ele deve ser geralmente usado apenas para fins de diagnóstico, não como um brinquedinho do tipo "Ei, veja que ferramentas legais eu tenho nas minhas janelas sobressalentes".

O root pode executar **top** com a opção **q** para elevá-lo ao mais alto nível de prioridade possível. Isso pode ser muito útil quando tentamos rastrear um processo que já subjugou o sistema.

4.9 Processos Descontrolados

Veja o Capítulo 26 para obter maiores informações sobre processos descontrolados.

Os processos descontrolados vêm em duas variantes: processos de usuário que usam quantidades excessivas de um recurso de sistema como tempo de CPU ou espaço em disco e processos de sistema que de repente se enfurecem e demonstram um comportamento selvagem. O primeiro tipo de descontrole não é necessariamente um defeito; pode ser simplesmente que o processo seja um comilão de recursos. Os processos de sistema supostamente sempre devem se comportar de maneira razoável.

Podemos identificar processos que usam tempo de CPU em excesso observando a saída gerada pelo comando **top**. Se estiver evidente que um processo de usuário está consumindo mais CPU do que poderia se esperar de forma razoável, investigue este processo. O passo número um é entrar em contato com o proprietário do processo e perguntar a ele o que está acontecendo. Se o proprietário não puder ser localizado, você terá de verificar por conta própria. Embora normalmente devamos evitar a investigação dos diretórios dos usuários, é aceitável quando estamos tentando descobrir a fonte de um processo descontrolado para descobrir o que está acontecendo.

Há duas razões para descobrir o que um processo está tentando fazer antes de manipulá-lo. Em primeiro lugar, o processo pode realmente ser legítimo e importante para o usuário. Não é razoável extinguir processos a esmo apenas porque por acaso eles usam um bocado de recursos de CPU. Em segundo lugar, o processo pode ser mal-intencionado ou destrutivo. Nesse caso, você tem de descobrir o que o processo está fazendo (por exemplo, quebrando senhas) para que você possa corrigir o estrago.

www.snow666.blogspot.com
Snow666

Se a razão para a existência de um processo descontrolado não puder ser determinada, suspenda-o com um sinal STOP e envie e-mail para o proprietário dele explicando o que aconteceu. O processo poderá ser reiniciado posteriormente emitindo-se um sinal CONT. Esteja ciente de que alguns processos podem ser arruinados após uma dormência muito longa, de modo que esse procedimento nem sempre é inteiramente benigno. Por exemplo, um processo poderia despertar e descobrir que algumas de suas conexões de rede foram desfeitas.

Se um processo usa uma quantidade excessiva de CPU, mas parece realizar algo razoável e funcionar corretamente, você deve emitir um **renice** passando-o para um valor nice maior (ou seja, prioridade menor) e solicitar ao proprietário para fazer o procedimento de nice futuramente.

Processos descontrolados que geram saídas podem esgotar um sistema de arquivos inteiro, provocando numerosos problemas. Quando um sistema de arquivos é esgotado, um monte de mensagens é registrado na console e tentativas de se gravar no sistema de arquivos produzem mensagens de erro.

A primeira coisa a fazer nessa situação é interromper o processo que está esgotando o disco. Se você vem mantendo uma reserva de espaço em disco razoável, pode estar quase certo de que algo está errado quando ele repentinamente se esgota. Não há nenhum comando análogo a **ps** que irá informá-lo sobre quem está consumindo espaço em disco da forma mais rápida, porém várias ferramentas são capazes de identificar arquivos que estão abertos atualmente e os processos que os estão usando. Veja as informações sobre **fuser** e **lsof** na seção 5.2, Capítulo 5, para obter maiores informações.

Talvez você queira suspender todos os processos que parecem suspeitos até localizar aquele que está provocando o problema, porém lembre-se de reiniciar os inocentes quando tiver terminado o serviço. Ao localizar o processo danoso, elimine os arquivos que ele estava criando.

Um velho e bem conhecido truque é iniciar um loop infinito no shell que faz o seguinte:

```
while 1
     mkdir adir
     cd adir
     touch afile
end
```

Esse programa ocasionalmente se apresenta executando em um login desprotegido ou em um terminal que foi deixado logado. Isso não consome tanto espaço em disco, porém ele esgota a tabela inode do sistema de arquivos e impede que outros usuários criem outros arquivos. Não há muito que possamos fazer, exceto limpar os seus efeitos e alertar os usuários para protegerem suas contas. Pelo fato de a árvore de diretórios que é deixada para trás por essa pequena "jóia" ser normalmente muito grande para ser tratada por **rm -r**, talvez você tenha de criar um script que vá até as partes mais internas da árvore e depois eliminar os diretórios à medida que ele for recuando pela árvore.

Se o problema ocorrer em **/tmp** e você configurou **/tmp** como um sistema de arquivos distinto, poderá reinicializar **/tmp** com **newfs** em vez de tentar eliminar arquivos individuais. Veja o Capítulo 8 para maiores informações sobre gerenciamento de sistemas de arquivos.

4.10 Exercícios

E4.1 Explique a relação existente entre o UID de um arquivo e os UIDs real e efetivo de um processo em execução. Além de controle de acesso a arquivos, qual é o propósito do UID efetivo de um processo?

E4.2 Suponha que um usuário em sua instalação tenha iniciado um processo de longa duração que está consumindo uma fração significativa dos recursos de uma máquina.

a) *Como* você reconheceria um processo que está exaurindo recursos?

b) Suponha que o processo de comportamento adverso possa ser legítimo e não deva ser extinto. Enumere os comandos que você poderia usar para colocá-los "na geladeira" (interrompê-lo temporariamente enquanto investiga).

c) Mais tarde, você descobre que o processo pertence a seu chefe e tem de continuar a ser processado. Enumere os comandos que você usaria para retomar a tarefa.

d) Alternativamente, suponha que o processo tenha de ser extinto. Que sinal você deveria emitir e por quê? E se você precisasse garantir que o processo fosse efetivamente extinto?

E4.3 Use **ps** e **top** com os flags apropriados para exibir todos os processos em andamento. Compare os resultados. Que coisas são mais fáceis de serem determinadas com **top** e quais com **ps**? Que nível de carga **top** coloca sobre o sistema?

E4.4 Localize um processo com estouro de memória (seu próprio programa ou **netscape** caso não tenha um à mão). Use **ps** ou **top** para monitorar o uso de memória do programa à medida que ele é executado.

Snow666

★ **E4.5** Desenhe uma figura representando a execução de um comando em um shell em execução. Com time (t) como eixo x, inicie com um simples retângulo representando o shell em execução em t=0. Suponha que o usuário digite um comando para o shell. Mostre em t=1 o resultado da execução da chamada de sistema **fork**. A seguir ocorrem as chamadas de sistema **exec**; mostre em t=2 os efeitos dos processos envolvidos. O comando finalmente é encerrado quando ele chama a rotina **exit**; mostre o estado do sistema em t=3.

★ **E4.6** Use o comando **ps** e um script **awk** simples (exemplo dado abaixo) para determinar o RSS e o VSZ total de seu sistema. Como estes números se relacionam com a quantidade de memória física e espaço de swap disponíveis? Escolha nomes de variáveis razoáveis para sua versão do script. (Este exemplo não é funcional, ou seja, errado. Use-o apenas como um exemplo.)

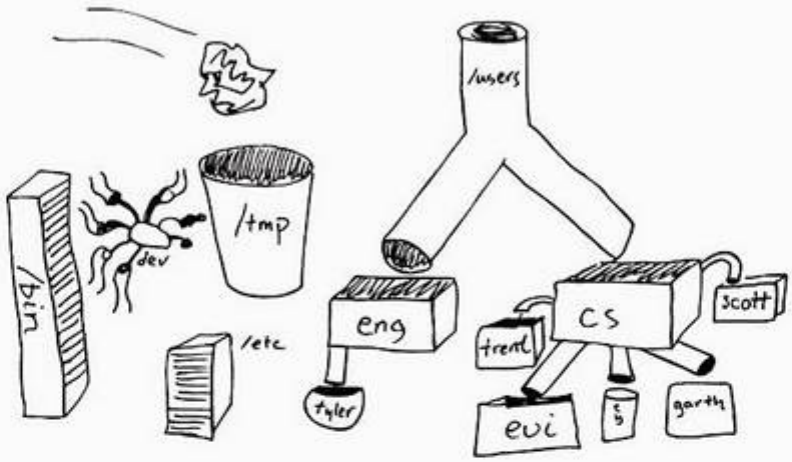
```
BEGIN {
    sum5=0
    sum6=0
}
{
    sum5 +=$5
    sum6 +=$6
}
END {
    printf("sum5=%d sum6=%d\n", sum5, sum6);
}
```

Uma linha de comando com o comando **ps** e o script **awk** acima seria:

```
% ps aux | awk 'BEGIN { sum5 = 0; sum6 = 0;} {sum5 +=$5;sum6 +=$6;} END {printf("sum5=%d
sum6=%d\n",sum5,sum6);}'
```

Snow666

5 O Sistema de Arquivos



Responda rapidamente: qual dos itens abaixo você esperaria encontrar num "sistema de arquivos"?

- Processos
- Portas seriais
- Canais de comunicação entre processos

Se o sistema for Linux, a resposta seria "todas as anteriores". E, logicamente, você poderá encontrar alguns arquivos lá também.

Embora o objetivo básico de um sistema de arquivos seja o de representar e organizar os recursos de armazenamento de um sistema, os programadores têm sido ávidos por evitar reinventar a roda quando se trata de gerenciar outros tipos de objetos. Freqüentemente, tem sido provado ser natural e conveniente associar tais objetos no espaço de nomes do sistema de arquivos. Há algumas vantagens para essa unificação (interface de programação consistente, fácil acesso no shell) e algumas desvantagens (implementações de sistema de arquivos tipo Frankenstein), mas goste ou não dele, esta é a maneira do UNIX (e, conseqüentemente, do Linux também).

O sistema de arquivos pode ser imaginado como compreendendo quatro componentes principais:

- Um espaço de nomes — uma maneira de dar nomes às coisas e organizá-las de forma hierárquica.
- Uma API[1] — um conjunto de chamadas de sistema para navegar e manipular objetos.
- Um modelo de segurança — um esquema para proteger, ocultar e compartilhar coisas.
- Uma implementação — software que amarra o modelo lógico ao hardware em si.

O Linux define uma interface abstrata em nível de kernel que acomoda vários sistemas de arquivos auxiliares distintos. Algumas partes da árvore de arquivos são manipuladas por implementações tradicionais baseadas em disco; outras são tratadas por drivers distintos dentro do kernel. Por exemplo, sistemas de arquivos NFS são manipulados por um driver que encaminha as operações solicitadas a um servidor ou a um outro computador.

Infelizmente, os limites arquitetônicos não são definidos claramente e há um número razoável de casos especiais. Por exemplo, arquivos de dispositivos fornecem uma maneira para os programas se comunicarem com drivers dentro do kernel. Eles não são realmente arquivos de dados, porém são manipulados pelo driver de sistema de arqui-

1. API (Application Programming Interface, interface de programas aplicativos), um termo genérico para o conjunto de rotinas que uma biblioteca, sistema operacional ou pacote de software fornece aos programadores para chamada.

Snow666

vos básico e suas características são armazenadas no disco. Talvez os detalhes sejam um tanto diferentes caso o sistema de arquivos fosse reimplementado à luz da experiência das últimas décadas.

Vá ao site www.namesys.com para obter maiores informações sobre ReiserFS.

Um outro fator complicador (mas em última instância benéfico) é o fato de o Linux suportar mais do que um sistema de arquivos baseado em disco. Além do sistema de arquivos ext2fs que serve como padrão para a maioria dos sistemas, há implementações que apresentam como característica um maior desempenho ou uma maior facilidade de recuperação após falhas, como o ReiserFS, JFS da IBM, XFS da SGI e o vindouro ext3fs (o ext3fs já está presente nos kernels da série 2.4 que vêm com as principais distribuições atuais). Há também muitas implementações de sistema de arquivos estrangeiros como o sistema de arquivos FAT usado no Microsoft Windows e o sistema de arquivos ISO-9660 usado em CD-ROMs. O Linux suporta mais tipos de sistemas de arquivos do que qualquer outra variante do UNIX. Sua ampla gama de opções nos dá grande flexibilidade e torna mais fácil compartilhar arquivos com outros sistemas.

O sistema de arquivos é um tópico rico que abordamos sob diversos ângulos. Este capítulo informa onde encontrar coisas sobre o seu sistema e descreve as características dos arquivos, os significados de bits de permissão e o emprego de alguns comandos básicos que visualizam e configuram atributos. O Capítulo 8, *Adicionando um Disco*, é onde você encontrará os tópicos mais técnicos sobre sistemas de arquivos como partição de discos. O Capítulo 17, *O Sistema de Arquivos de Rede*, descreve os sistemas de compartilhamento de arquivos que são comumente usados com o Linux.

5.1 Nomes de Caminhos

O sistema de arquivos é apresentado como uma única hierarquia[2] unificada que se inicia no diretório **/** e prossegue abaixo até um número arbitrário de subdiretórios. **/** também é chamado de diretório-raiz.

A lista de diretórios que deve ser percorrida para localizar um determinado arquivo, juntamente com o seu nome de arquivo, forma um "nome de caminho". Os nomes de caminho podem ser tanto absolutos (**/tmp/foo**) como relativos (**book4/filesystem**). Os nomes de caminho são interpretados no diretório atual. Você pode estar acostumado a pensar no diretório atual como um recurso do shell, mas cada processo possui um diretório atual.

Os termos *arquivo*, *nome de arquivo*, *nome de caminho* e *caminho* são mais ou menos intercambiáveis (ou, pelo menos, nós os usamos de forma intercambiável neste livro). *Nome de arquivo* e *caminho* podem ser usados tanto para caminhos absolutos quanto para caminhos relativos; *nome de caminho* geralmente sugere um caminho absoluto.

O sistema de arquivos pode ser arbitrariamente "profundo". Entretanto, cada componente de um nome de caminho tem de ter um nome que não supere 255 caracteres, e um caminho único não pode conter mais do que 4.095 caracteres. Para acessar um arquivo com um nome de caminho superior a isso, você tem de usar **cd** para um diretório intermediário e usar um nome de caminho relativo.[3]

Não há, fundamentalmente, nenhuma restrição na atribuição de nomes de arquivos e diretórios, exceto pelo fato de os nomes serem limitados em comprimento e não poderem conter o caractere "/" ou nulos. Em particular, são permitidos espaços ... mais ou menos. Pelo fato da longa tradição do UNIX em separar argumentos de linha de comando com espaços em branco, softwares mais antigos tendem a parar quando encontram espaços dentro de nomes de arquivos.

Dada a grande quantidade de compartilhamento de arquivos entre diferentes tipos de sistemas existentes hoje em dia, não é mais seguro supor que os nomes de arquivos não conterão espaços. Mesmo que você não compartilhe arquivos com Macs e PCs, há uma série de usuários que têm o hábito de usar espaços em nomes de arquivos.[4] Qualquer software que você crie com o sistema de arquivos deve estar preparado para lidar com espaços.

No shell e em scripts, nomes de arquivos contendo espaços simplesmente precisam ser colocados entre aspas para manter os seus componentes juntos. Por exemplo, o comando

```
% more "Meu excelente arquivo.txt"
```

manteria **Meu excelente arquivo.txt** como um único argumento de **more**.

---

2. O sistema de hierarquia única difere daquele usado pelo Windows que conserva o conceito de espaços de nomes específicos a discos.
3. No caso de não ter ficado claro: a maioria dos formatos de sistemas de arquivos de disco não impõe por si só um limite em relação ao comprimento total de nomes de arquivos. Entretanto, as chamadas de sistema que acessam o sistema de arquivos não permitem que seus argumentos string sejam mais longos do que 4.095 caracteres.
4. De forma irritante, até mesmo alguns projetos de fonte aberto começaram a usar espaços. Por exemplo, o KDE e o GNOME criam arquivos e links simbólicos contendo espaços em seus nomes. Um de nossos revisores escreveu: "A maioria dos sistemas operacionais usa um espaço como delimitador. Usar espaços em nomes de arquivos é *BURRICE* — até mesmo o Windows sofre deste problema na linha de comandos.

www.snow666.blogspot.com
Snow666

Alguns shells incluem as quotas "automagicamente", de modo que se você estiver utilizando o **bash**, que é o shell comum na maioria das distribuições do Linux, o comando poderia ser

```
%% more Meu excelente arquivo.txt
```

5.2 Montando e Desmontando Sistemas de Arquivos

O sistema de arquivos é composto por pequenos blocos — também chamados de sistemas de arquivos —, cada um dos quais formado por um diretório e seus subdiretórios e arquivos. É normalmente evidente, a partir do contexto, que tipo de "sistema de arquivos" está sendo discutido, porém, para fins de clareza, usaremos o termo "árvore de arquivos" para nos referirmos ao layout geral do sistema de arquivos e reservaremos o termo "sistema de arquivos" para os blocos anexados à árvore.

A maioria dos sistemas de arquivos são partições de disco, porém, conforme mencionado anteriormente, eles podem ser qualquer coisa desde que obedeçam à API apropriada: servidores de arquivos de rede, componentes de kernel, emuladores de disco baseado em memória etc. O Linux tem até um sistema de arquivos "loopback" engenhoso que nos permite montar arquivos individuais como se fossem dispositivos distintos.

Os sistemas de arquivos são anexados à árvore com o comando **mount**. **mount** associa um diretório dentro da árvore de arquivos existente, chamado de ponto de montagem, à raiz do sistema de arquivos recém-anexado. O conteúdo anterior do ponto de montagem se torna inacessível enquanto um outro sistema de arquivos esteja montado lá. Entretanto, os pontos de montagem são normalmente diretórios vazios.

Por exemplo,

```
# mount /dev/hda4 /users
```

instalaria o sistema de arquivos armazenado na partição de disco representada por **/dev/hda4** sob o caminho **/users**. Poderíamos então usar **ls /users** para ver o conteúdo do sistema de arquivos.

Uma lista dos sistemas de arquivos montados costumeiramente num determinado sistema é mantida no arquivo **/etc/fstab**. A informação contida nesse arquivo permite que os sistemas de arquivos sejam checados (**fsck -A**) e montados (**mount -a**) automaticamente no momento da inicialização. Ela também serve como documentação para o layout da "árvore de arquivos" no disco e permite utilizar comandos abreviados, tais como **mount /usr** (a localização do sistema de arquivos a ser montado é procurada no arquivo **fstab**). Veja no Capítulo 8, seção 8.4, uma discussão completa sobre o arquivo **fstab**.

Os sistemas de arquivos são desanexados através do comando **umount**. Não podemos desmontar um sistema de arquivos que está "ocupado"; não deve haver nenhum arquivo ou processo aberto cujos diretórios atuais estejam lá localizados, e se o sistema de arquivos contiver programas executáveis, eles não podem estar em execução.

Se o kernel reclamar que um sistema de arquivos que você está tentando desmontar está ocupado, você não poderá executar **fuser** para descobrir o porquê. Quando invocado com flags **-mv** e um ponto de montagem, **fuser** exibe cada processo que está usando um arquivo ou diretório neste sistema de arquivos:

```
% fuser -mv /usr
         USR   PID   ACCESS   COMMAND
/usr     root  444   ...m        atd
         root  499   ...m        sshd
         root  520   ...m        lpd
         ...
```

Os códigos de letras na coluna ACCESS mostram o que cada processo está fazendo para interferir com sua tentativa de desmontagem. A Tabela 5.1 descreve o significado de cada código.

Tabela 5.1 Códigos de atividade exibidos por fuser.

Código	Significado
f	O processo possui um arquivo aberto para leitura ou gravação.
c	O diretório atual do processo se encontra no sistema de arquivos.
e	O processo está executando um arquivo no momento.
r	O diretório-raiz do processo (configurado através de **chroot**) se encontra no sistema de arquivos.
m	O processo associou um arquivo ou biblioteca compartilhada (normalmente um executável inativo).

Snow666

Para determinar exatamente quais são os processos transgressores, basta executar **ps** com a lista de PIDs retornadas por **fuser**. Por exemplo:

```
% ps -fp "444 499 520"
     UID   PID   PPID  C  STIME  TTY       TIME      CMD
  daemon   444   1     0  April  question  00:00:00  /usr/sbin/atd
    Root   499   1     0  April  question  00:00:23  /usr/sbin/sshd
      lp   520   1     0  April  question  00:00:00  [lpd]
```

As aspas forçam o shell a passar a lista de PIDs para **ps** como um argumento único.

fuser também pode relatar o uso de arquivos específicos (ao contrário de sistemas de arquivos inteiros); a sintaxe é **fuser -v** *nome_do_arquivo*. **fuser** também aceita a opção **-k** para extinguir (ou enviar um sinal para) cada um dos processos transgressores. Perigoso — e você tem de ser o root (ou usar **sudo**, veja o Capítulo 3).

Uma alternativa para **fuser** é o utilitário gratuito **lsof** de Vic Abell da Purdue University. A saída de **lsof** é um tanto mais prolixa do que **fuser**. **lsof** se encontra disponível em

`ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/`

lsof é disponível para o Red Hat e SuSE no formato RPM no site www.redhat.com. Ele pode ser obtido em sistemas Debian através de **apt-get**. Entretanto, esses dois empacotamentos Linux (rpm e deb) estavam um tanto defasados na época em que este livro foi escrito.

5.3 A Organização da Árvore de Arquivos

Os sistemas de arquivos na família UNIX jamais foram bem organizados. Várias convenções de nomes incompatíveis são usadas ao mesmo tempo e diferentes tipos de arquivos são espalhados aleatoriamente pelo espaço de nomes. Em muitos casos, os arquivos são divididos por função, e não pela probabilidade com que eles mudam, dificultando a atualização do sistema operacional. O diretório **/etc**, por exemplo, contém alguns arquivos que jamais são personalizados e alguns deles são inteiramente locais. Como saber quais arquivos preservar durante a atualização? Bem, você simplesmente tem de saber ...

Inovações como **/var** ajudaram a solucionar alguns problemas, porém a maioria dos sistemas ainda é uma perfeita bagunça. Não obstante, há um local culturalmente correto para tudo. É particularmente importante não bagunçar a estrutura-padrão da árvore de arquivos sob o Linux, pois os pacotes de software e suas ferramentas de instalação normalmente fazem suposições abrangentes em relação à localização dos arquivos (assim como fazem outros administradores de sistemas!).

Consulte o Capítulo 12 para obter maiores informações sobre a configuração do kernel.

O sistema de arquivos raiz inclui o diretório-raiz e um conjunto mínimo de arquivos e subdiretórios. O arquivo contendo o kernel reside dentro do sistema de arquivos raiz no diretório **/boot**; ele é normalmente chamado de (ou associado a) **vmlinuz**. Também parte do sistema de arquivos raiz são **/dev** para os arquivos de dispositivos (exceto **/dev/pts**, que é montado separadamente), **/etc** para sistemas de arquivos críticos, **/sbin** e **/bin** para utilitários importantes e, algumas vezes, **/tmp** para arquivos temporários.

Consulte o Capítulo 8, seção 8.3, para obter alguns motivos pelos quais o particionamento pode ser desejável e algumas regras práticas para orientá-lo.

Os diretórios **/usr** e **/var** também são de grande importância. **/usr** é onde a maioria dos programas-padrão é mantida, juntamente com vários outros brindes como manuais on-line e a maioria das bibliotecas. Não é estritamente necessário que **/usr** seja um sistema de arquivos separado, mas para conveniência administrativa geralmente é. Tanto **/usr** quanto **/var** têm de estar disponíveis para habilitar o sistema a trilhar todo o caminho até o modo multiusuário.

/var abriga os diretórios spool, arquivos de log, informações contábeis e vários outros itens que crescem ou modificam rapidamente e variam em cada host. Já que **/var** contém arquivos de log, que estão aptos a crescer em tempos de problemas, é uma boa idéia colocá-los em seu próprio sistema de arquivos se isto for prático.

Os diretórios home dos usuários normalmente são mantidos num sistema de arquivos separado, usualmente montado no diretório-raiz. Sistemas de arquivos distintos também podem ser usados para armazenar itens volumosos como bancos de dados e bibliotecas de código-fonte.

Alguns dos diretórios-padrão mais importantes são apresentados na Tabela 5.2.

www.snow666.blogspot.com
Snow666

Tabela 5.2 Diretórios-padrão e seu conteúdo.

Nome do caminho	Conteúdo
/bin	Comandos necessários para a mínima operação do sistema.
/boot	Kernel e arquivos necessários para carregar o kernel.
/dev	Entradas de dispositivos para terminais, discos, modems etc.
/etc	Arquivos de configuração e inicialização críticos.
/lib	Bibliotecas e partes do compilador C.
/opt	Pacotes de software aplicativo opcionais e agregáveis.
/proc	Imagens de todos os processos em andamento.
/root	Diretório de usuário do superusuário (normalmente apenas /).
/sbin	Comandos para inicialização, reparos ou recuperação do sistema.
/tmp	Arquivos temporários que desaparecem entre as reinicializações.
/usr	Hierarquia de arquivos e comandos secundários.
/usr/bin	A maioria dos arquivos executáveis e comandos.
/usr/include	Arquivos de cabeçalho para programas em C.
/usr/lib	Bibliotecas; também, arquivos de programas suporte para programas-padrão.
/usr/local	Software local (coisas que instalamos).
/usr/local/bin	Executáveis locais.
/usr/local/etc	Arquivos de configuração e comandos locais.
/usr/local/lib	Arquivos de suporte locais.
/usr/local/sbin	Comandos locais de manutenção de sistema linkados estaticamente.
/usr/local/src	Código-fonte para **/usr/local/***.
/usr/sbin	Comandos menos essenciais para administração e reparo de sistema.
/usr/share	Itens que podem ser comuns a vários sistemas (somente para leitura).
/usr/share/man	Páginas de manuais on-line.
/usr/src	Código-fonte para pacotes de software (não locais).
/usr/src/linux	Área de trabalho de construção de kernel, arquivos de configuração.
/var	Arquivos de configuração e dados específicos ao sistema.
/var/adm	Vários: logs, registros de configuração de sistema, pequenas coisas estranhas para administração do sistema.
/var/log	Vários arquivos de log de sistema.
/var/spool	Diretórios de spool para impressão, e-mails etc.
/var/tmp	Mais espaço temporário (preservado entre reinicializações).

O evolucionário FHS (Filesystem Hierarchy Standard, padrão de hierarquia de sistemas de arquivos) [www.pathname.org/fhs] tenta codificar, racionalizar e explicar os diretórios-padrão. É um excelente recurso para consulta quando tentamos descobrir onde colocar algo. Discutimos algumas considerações adicionais no projeto de hierarquias locais a partir da página 544.

5.4 Tipos de Arquivos

O Linux define sete tipos de arquivos. Mesmo quando os desenvolvedores adicionam alguma coisa nova e maravilhosa à árvore de arquivos (como as informações de processo listadas em **/proc**), ela ainda tem de ser feita de maneira a se parecer com um dos sete tipos a seguir:

- Arquivos regulares
- Diretórios
- Arquivos de dispositivos de caracteres
- Arquivos de dispositivos de blocos
- Sockets de domínio local

Snow666

- Pipes com nome (FIFOs)
- Links simbólicos

Podemos determinar o tipo de arquivo existente através de **ls -ld**. O primeiro caractere da saída gerada por **ls** codifica o tipo. O exemplo a seguir demonstra que **/usr/include** é um diretório:

```
% ls -ld /usr/include
(d)rwxr-xr-x 27 root     root     4096  Jul 15 20:57   /usr/include
```

ls usa os códigos apresentados na Tabela 5.3 para representar os diversos tipos de arquivos.

Tabela 5.3 Codificação de tipos de arquivos usados por ls.

Tipo de arquivo	Símbolo	Criado por	Eliminado por
Arquivo comum	-	editores, **cp** etc.	**rm**
Diretório	d	**mkdir**	**rmdir, rm -r**
Arquivo de dispositivos de caracteres	c	**mknod**	**rm**
Arquivo de dispositivos de blocos	b	**mknod**	**rm**
Pipe com nome	p	**mknod**	**rm**
Link simbólico	l	**ln -s**	**rm**

Como pode ser observado na Tabela 5.3, **rm** é a ferramenta universal para eliminar arquivos que não queremos mais. Mas como iríamos eliminar um determinado arquivo, digamos, **-f**? É um nome de arquivo perfeitamente legítimo na maioria dos sistemas de arquivos, porém **rm -f** não funciona, pois **-f** é interpretado como um flag **rm**. A resposta é referir-se ao arquivo por um nome de caminho completo (como **./-f**) ou para usar o argumento -- de **rm** para informá-lo de que tudo o que vier a seguir é um nome de arquivo, e não uma opção (por exemplo, **rm -- -f**).

Os nomes de arquivo contendo caracteres de controle apresentam problema similar, já que pode ser difícil ou impossível reproduzir estes nomes pelo teclado. Nesse caso, podemos usar globalização de shell (concordância de padrões) para identificar os arquivos a serem eliminados. Ao usar concordância de padrões, é uma boa idéia se acostumar com o emprego da opção **-i** de **rm** para fazer com que **rm** confirme a eliminação de cada arquivo. Esse recurso o protegerá contra eliminação de qualquer arquivo "bom" que seu padrão inadvertidamente venha a concordar. Por exemplo, para eliminarmos um arquivo chamado **foo**<Control-D>**bar**, usaríamos

```
% ls
foo?bar     foose     kde-root

% rm -i foo*
rm: remove 'foo/004bar'? y
rm: remove 'foose'? n
```

Note que **ls** exibe o caractere de controle como um ponto de interrogação, o que pode ser um tanto enganoso. Se você não se lembrar que **?** é um caractere de concordância de padrão de shell e tentar **rm foo?bar**, corre o risco de eliminar mais do que um arquivo (embora não neste exemplo). **-i** é seu amigo!

Para eliminar os arquivos de nomes mais complicados, talvez seja necessário executar **rm -i ***.

Arquivos regulares

Um arquivo regular é apenas uma coletânea de bytes; o Linux não impõe nenhuma estrutura sobre seu conteúdo. Os arquivos de texto, arquivos de dados, programas executáveis e bibliotecas compartilhadas são armazenados na forma de arquivos regulares. É permitido acesso seqüencial, bem como aleatório.

Diretórios

Um diretório contém referências com nomes para outros arquivos. Podemos criar diretórios através de **mkdir** e eliminá-los através de **rmdir** caso estejam vazios. Podemos eliminar diretórios não vazios com **rm -r**.

As entradas especiais "**.**" e "**..**" referem-se ao diretório em si e ao seu diretório-pai; elas não podem ser eliminadas. Já que o diretório-raiz não possui nenhum diretório-pai, o caminho "**/..**" é equivalente ao caminho "**/.**" (e ambos são equivalentes a /).

www.snow666.blogspot.com
Snow666

O nome de um arquivo é realmente armazenado dentro de seu diretório-pai, não com o arquivo em si. De fato, mais do que um diretório (ou mais do que uma entrada num único diretório) pode-se referir a um arquivo por vez, e as referências podem ter nomes diferentes. Tal arranjo cria a ilusão de que um arquivo existe em mais de um lugar ao mesmo tempo.

Essas referências adicionais ("links") são indistinguíveis do arquivo original; no que diz respeito ao Linux, elas são equivalentes. O Linux mantém uma contagem do número de links que apontam para cada arquivo e não libera os blocos de dados do arquivo até que seu último link tenha sido eliminado. Os links não podem cruzar os limites de sistemas de arquivos.

Referências desse tipo são normalmente chamadas de "hard links" nos dias de hoje para distingui-las de links simbólicos, que são descritos a seguir. Criamos hard links com **ln** e os eliminamos com **rm**.

É fácil se lembrar da sintaxe de **ln** caso você tenha em mente que ela espelha aquela de **cp**. O comando **cp oldfile newfile** cria uma cópia de **oldfile** chamada **newfile**, e **ln oldfile newfile** torna o nome **newfile** uma referência adicional a **oldfile**.

É importante compreender que os hard links não são um tipo distinto de arquivo. Em vez de definir uma "coisa" distinta chamada hard link, o sistema de arquivos simplesmente permite que mais de uma entrada de diretório aponte para um determinado arquivo. Além do conteúdo do arquivo, os atributos subjacentes do arquivo (como as propriedades e permissões) também são compartilhados entre links.

Arquivos de dispositivos de blocos e caracteres

Veja o Capítulo 12 para obter maiores informações sobre dispositivos e drivers.

Os arquivos de dispositivos permitem que os programas se comuniquem com o hardware e periféricos do sistema. Quando o kernel é configurado, os módulos que sabem como se comunicar com cada um dos dispositivos do sistema são associados.[5] O módulo para um determinado dispositivo, chamado de driver de dispositivo, cuida dos confusos detalhes do gerenciamento de dispositivos.

Os drivers de dispositivos apresentam uma interface de comunicação padrão que se parece com um arquivo comum. Quando o kernel recebe uma solicitação que se refere a um arquivo de dispositivo de blocos ou caracteres, ele simplesmente passa a solicitação ao driver de dispositivo apropriado. Entretanto, é importante distinguir *arquivos* de *drivers* de *arquivos de* dispositivos. Os arquivos são simplesmente pontos de encontro usados para se comunicarem com os drivers. Eles não são os drivers em si.

Os arquivos de dispositivos de caracteres permitem que seus drivers associados façam seus próprios bufferings de entrada e saída. Os arquivos de dispositivos de blocos são usados por drivers que manipulam I/O em grandes blocos e querem que o kernel execute buffering para eles. Alguns tipos de hardware como, por exemplo, os discos rígidos, podem ser representados tanto por arquivos de dispositivos de caracteres como de blocos. (Como saber que versão usar para uma dada finalidade? Infelizmente, não há nenhuma regra prática — temos de memorizar os casos comuns ou pesquisá-los.)

Os arquivos de dispositivos são caracterizados por dois números, chamados de números de dispositivos principal e secundário. O número de dispositivo principal informa o kernel à qual driver o arquivo se refere e o número de dispositivo secundário informa ao driver que unidade física tratar. Por exemplo, o número de dispositivo principal 6 num sistema Linux indica o driver de porta paralela. A primeira porta paralela (**/dev/lp0**) teria o número de dispositivo principal 6 e o número de dispositivo secundário seria 0.

Alguns drivers de dispositivos usam o número de dispositivo secundário de uma maneira não padronizada. Por exemplo, drivers de unidade de fita usam o número de dispositivo secundário para determinar se a fita deve ser rebobinada quando o arquivo de dispositivo é fechado. Em alguns casos, mais do que um driver pode se referir ao mesmo dispositivo físico. Por exemplo, os arquivos **/dev/ttyS0** e **/dev/cua0** possuem números de dispositivos principal diferentes, mas se referem à mesma porta serial. Eles implementam um esquema de conexão para modems bidirecionais.

Podemos criar arquivos de dispositivos com **mknod** e eliminá-los com **rm**. A maioria dos sistemas fornece um script chamado **/dev/MAKEDEV** que cria os conjuntos apropriados de arquivos de dispositivo para dispositivos comuns. No SuSE, este programa não é instalado por padrão. Veja o Capítulo 12, *Drivers e o Kernel*, para obter maiores informações.

5. Esses módulos também podem ser carregados dinamicamente pelo kernel.

www.snow666.blogspot.com
Snow666

Sockets de domínio local

Sockets são conexões entre processos que permitem a eles se comunicarem de uma maneira sadia. O Linux fornece vários tipos de sockets, a maioria dos quais envolve o emprego de uma rede. Sockets de domínio local são acessíveis somente do host local e referidos através de um objeto do sistema de arquivos em vez de uma porta de rede. Eles são conhecidos algumas vezes como "sockets de domínio UNIX".

Veja o o Capítulo 11 para obter maiores informações sobre syslog.

Embora os arquivos de socket sejam visíveis para outros processos na forma de entradas de diretório, eles não podem ser lidos ou gravados por processos não envolvidos na conexão. Alguns recursos-padrão que usam sockets de domínio local são o sistema de impressão, o sistema X Window e syslog.

Sockets de domínio local são criados com a chamada de sistema **socket** e podem ser eliminados com o comando **rm** ou com a chamada de sistema **unlink** assim que eles não tiverem mais usuários.

Pipes com nomes

Assim como os sockets de domínio local, os pipes com nomes permitem a comunicação entre dois processos rodando no mesmo host. Eles também são conhecidos como "arquivos FIFO" (FIFO é a abreviatura para "first in, first out", ou seja, "o primeiro a entrar é o último a sair"). Podemos criar pipes com nomes via **mknod** e eliminá-los com **rm**.

Como acontece com sockets de domínio local, as instâncias do mundo real de pipes com nomes são poucas e distantes. Eles raramente, se nunca, requerem intervenção administrativa.

Links simbólicos

Um link simbólico ou "soft" aponta para um arquivo pelo nome. Quando o kernel vem após um link simbólico no curso da pesquisa de um nome de caminho, ele redireciona sua atenção para o nome de caminho armazenado como conteúdo do link. A diferença entre hard links e links simbólicos é que o hard link é uma referência direta, ao passo que um link simbólico é uma referência por nome; links simbólicos são distintos de arquivos para os quais eles apontam.

Criamos links simbólicos com **ln -s** e os eliminamos com **rm**. Já que podem conter caminhos arbitrários, podem se referir a arquivos em outros sistemas de arquivos ou a arquivos inexistentes. Links simbólicos múltiplos também podem formar um loop.

Um link simbólico pode conter tanto um caminho absoluto como um caminho relativo. Por exemplo,

```
# ln -s ncursers/term.h /usr/include/term.h
```

associa **/usr/include/term.h** a **/usr/include/ncurses/term.h** com um caminho relativo (o comando acima foi executado de dentro do diretório /usr/include); ele cria o link simbólico **/usr/include/term.h** com um destino de "**ncurses/term.h**". O diretório inteiro **/usr/include** poderia ser movido para algum outro ponto sem fazer com que o link simbólico pare de funcionar (não que mover o diretório seja recomendável).

É um erro comum pensar que o primeiro argumento para **ln -s** tenha algo a ver com seu diretório de trabalho atual. Ele *não* é interpretado como um nome de arquivo por **ln**; ele é simplesmente usado literalmente como destino do link simbólico.

5.5 Atributos de Arquivos

Todo arquivo possui um conjunto de bits de permissão que controlam quem pode ler, gravar e executar o conteúdo de um arquivo. Juntos com três outros bits que afetam primariamente a operação dos programas executáveis, esses bits constituem o "modo" do arquivo.

Os 12 bits de modo são armazenados juntos com quatro bits de informações de tipos de arquivo. Os quatro bits de tipo de arquivo são configurados quando o arquivo é criado pela primeira vez e não podem ser modificados, porém o proprietário do arquivo e o superusuário podem modificar os 12 bits de modo usando o comando **chmod** (change mode). Use **ls -l** (ou **ls -ld** para um diretório) para inspecionar os valores destes bits. Um exemplo é dado na seção *Visualizando atributos de arquivos*.

Os bits setuid e setgid

Os bits com valores octais 4000 e 2000 são os bits setuid e setgid. Estes bits permitem aos programas acessarem arquivos e processos que de outro modo estariam fora do alcance do usuário que os executa. O mecanismo setuid/setgid para executáveis é descrito na página 33.

Quando configurado num diretório, o bit setgid faz com que arquivos recém-criados dentro do diretório assumam a propriedade do grupo do diretório em vez do grupo-padrão do usuário que criou o arquivo. Essa convenção torna mais fácil compartilhar um diretório de arquivos entre vários usuários, desde que todos eles pertençam a um

Snow666

grupo comum. Essa interpretação do bit setgid não está relacionada ao seu significado quando configurada num arquivo executável, porém jamais há qualquer ambigüidade no que diz respeito a qual significado é apropriado.

Também podemos configurar o bit setgid em arquivos puros não executáveis para solicitar comportamento de bloqueio especial quando o arquivo é aberto. Entretanto, jamais vimos esse recurso em execução.

O bit sticky

O bit com valor octal 1000 é chamado de bit sticky. Ele foi de importância histórica como um modificador para arquivos executáveis nos primeiros sistemas UNIX. Entretanto, esse significado de bit sticky agora é obsoleto, e o Linux o ignora silenciosamente.

Se o bit sticky estiver configurado num diretório, o sistema de arquivos não irá permitir que eliminemos ou renomeemos um arquivo a menos que sejamos o proprietário do diretório, o proprietário do arquivo ou o superusuário. Ter permissão de gravação no diretório não é o suficiente. Essa convenção ajuda a tornar diretórios como **/tmp** um pouco mais privados e seguros.

Os bits de permissão

Os nove bits de permissão são usados para determinar que operações podem ser executadas num arquivo e por quem. O Linux não deixa que permissões sejam configuradas por usuário. Em vez disso, há conjuntos de permissões para o proprietário do arquivo, os proprietários do grupo do arquivo e todos os outros. Cada conjunto possui três bits: um bit de leitura, um bit de gravação e um bit de execução.

É conveniente discutir as permissões de arquivo em termos de números octais (base 8), pois cada dígito de um número octal representa 3 bits e há 3 bits em cada grupo de bits de permissão. Os três bits mais significativos (com valores 400, 200 e 100) controlam o acesso para o proprietário. Os próximos três (40, 20 e 10) controlam o acesso para o grupo. Os três últimos (4, 2 e 1) controlam o acesso para todo o restante ("o mundo"). Em cada trio , o bit alto é o bit de leitura, o bit intermediário é o bit de gravação e o bit baixo é o bit de execução.

Cada usuário atende somente a um desses três conjuntos de permissão. As permissões usadas são aquelas mais específicas. Por exemplo, o proprietário de um arquivo tem o acesso determinado pelos bits de permissão de usuário e jamais pelos bits de permissão de grupo. É possível para as categorias "outros" e "grupo" obterem mais acesso do que o proprietário, embora esta configuração raramente seja usada.

Num arquivo comum, o bit de leitura permite que o arquivo seja aberto e lido. O bit de escrita permite que o conteúdo do arquivo seja modificado ou truncado; entretanto, a capacidade de eliminar ou renomear (ou eliminar e depois recriar!) o arquivo é controlada pelas permissões em seu diretório-pai (pois é aí onde a associação [mapeamento] nome-espaço de dados é realmente armazenada).

Os bits de execução permitem que o arquivo seja executado. Há dois tipos de arquivos executáveis: binários, que a CPU executa diretamente, e scripts, que devem ser interpretados por um shell ou algum outro programa. Por convenção, os scripts começam com uma linha na forma

```
#!/bin/csh -f
```

que especifica um interpretador apropriado. Arquivos executáveis que não são binários e que não especifiquem um interpretador são considerados (pelo shell) com scripts **bash/sh**.[6]

Para um diretório, o bit de execução (normalmente chamado neste contexto de bit de "pesquisa" ou de "varredura") permite que o diretório seja introduzido ou passado enquanto um nome de caminho está sendo avaliado, mas não permite que o seu conteúdo seja listado. A combinação de bits de leitura e de execução permite que arquivos sejam criados, eliminados e renomeados dentro do diretório.

Visualizando atributos de arquivos

O sistema de arquivos mantém cerca de 40 blocos de informação distintos para cada arquivo, mas a maioria deles é útil apenas para o sistema de arquivos em si. Como administrador de sistemas, você deverá se preocupar principalmente com a contagem de links, proprietário, grupo, modo, tamanho, horário do último acesso, horário da última modificação e tipo. Você pode inspecionar todas eles através de **ls -l** (ou **ls -ld** para um diretório).

6. O kernel interpreta a sintaxe #! ("a coisa") e atua diretamente sobre ele. Entretanto, se o interpretador não for especificado completa e corretamente, o kernel se recusará a executar o arquivo. O shell faz então uma segunda tentativa de executar o script chamando **sh**.

www.snow666.blogspot.com
Snow666

Também é mantido um horário de mudança de atributo para cada arquivo. O nome convencional para esse horário (o "ctime", abreviatura de "change time") leva algumas pessoas a acreditar que ele seja o horário de criação do arquivo. Infelizmente, não é; ele apenas registra o momento em que os atributos do arquivo (proprietário, modo etc.) foram modificados pela última vez (e não o horário no qual o conteúdo do arquivo foi modificado).

Considere o exemplo a seguir:

```
% ls -l /bin/gzip
-rwxr-xr-x   3  root     root  51388 Jul 13 2000 /bin/gzip
```

O primeiro campo especifica o tipo e modo do arquivo. O primeiro caractere é um traço e, portanto, trata-se de um arquivo regular. (Veja a Tabela 5.3 na página 54 para saber sobre outros códigos.)

Os próximos nove caracteres desse campo são os três conjuntos de bits de permissão. A ordem dos grupos é proprietário-grupo-outros e a ordem dos bits dentro de cada grupo é leitura-gravação-execução. Embora esses bits tenham apenas valores binários, **ls** os exibe simbolicamente através das letras r, w e x de read, write e execute. Nesse caso, o proprietário tem todas as permissões sobre o arquivo e todos os demais somente têm permissão de leitura e de execução.

Se o bit setuid tivesse sido configurado, o x representando a permissão de execução do proprietário teria sido substituído por um `s` e, se o bit setgid tivesse sido ativado, o x para o grupo também teria sido substituído por um `s`. O último caractere de permissões (permissão de execução para "outros") é exibido como `t` se o bit sticky do arquivo estiver ativado. Se tanto o bit setuid/setgid ou o bit sticky estiverem ativados, mas o bit de execução correspondente não estiver, estes bits aparecerão como `S` ou `T`.

O próximo campo na listagem é a contagem de links para o arquivo. Nesse caso ele é 3, indicando que **/bin/gzip** é apenas um dos três nomes para esse arquivo (os outros são **/bin/gunzip** e **/bin/zcat**). Cada vez que é feito um hard link a um arquivo, a contagem é incrementada de 1 unidade.

Todos os diretórios têm ao menos dois hard links: o link do diretório-pai e o link do arquivo especial "." dentro do próprio diretório em si. Links simbólicos não afetam a contagem de links.

Os próximos dois campos na saída gerada por **ls** são o proprietário e proprietário de grupo do arquivo. Nesse exemplo, o proprietário do arquivo é root e o arquivo também pertence ao grupo chamado root. O sistema de arquivos na verdade armazena estes como números de ID's de usuário e de grupo, e não como nomes. Se as versões na forma de texto (nomes) não puderem ser determinadas, esses campos conterão números. Isso poderia acontecer caso o usuário ou o grupo que detém a propriedade do arquivo tivessem sido eliminados do arquivo **/etc/passwd** ou **/etc/group**. Isso também poderia indicar um problema com o seu banco de dados NIS (se você usar um); veja o Capítulo 18.

O próximo campo é o tamanho do arquivo em bytes. Este arquivo tem 51.388 bytes de comprimento, um pouco mais do que 50K.[7] A seguir temos a data da última modificação: 13 de julho de 2000. O último campo na listagem é o nome do arquivo, **/bin/gzip**.

A saída gerada por **ls** é ligeiramente diferente para um arquivo de dispositivo. Por exemplo:

```
% ls -l /dev/tty0
crw------   1 root   tty   4,   0 Aug 24  2000 /dev/tty0
```

A maioria dos campos é a mesma, mas em vez de um tamanho em bytes, **ls** exibe os números de dispositivo principal e secundário. **/dev/tty0** é a primeira console virtual, controlada pelo driver de dispositivo 4 (o driver de terminal).

Uma opção **ls** que é útil para examinar hard links é **-i**, que exibe o "número de inode" de cada arquivo. Sem entrar em muitos detalhes sobre a implementação do sistema de arquivos, diremos apenas que o número de inode é um índice numa tabela que enumera todos os arquivos no sistema de arquivos. Inodes são as "coisas" que são apontadas pelas entradas de diretórios; entradas que são hard links para o mesmo arquivo terão o mesmo inode. Para desco-

7. K significa kilo, um prefixo de medida que significa 1.000; entretanto, os sujeitos da computação o degeneraram seu significado, passando a 2^{10} ou 1.024. De forma similar, um megabyte em sistemas computacionais não é realmente um milhão de bytes mas sim 2^{20}, ou seja, 1.048.576 bytes. A Internaticonal Electrotechnical Commission está promovendo um novo conjunto de prefixos numéricos (como, por exemplo, kibi- e mebi-) que se baseiam explicitamente em potências de 2. Nesta altura, parece pouco provável que o emprego comum mudará. Para deixar a coisa ainda mais confusa, até mesmo as unidades de potência de 2 não são usadas de maneira consistente. RAM é denominada em potências de 2, mas a largura de banda é sempre uma potência de 10. O espaço de armazenamento é representado em unidades de potência de 10 pelos fabricantes e unidades de potência de 2 pelo restante.

Snow666

brir uma teia complexa de links, você precisará usar **ls -li** para exibir as contagens de links e números de inode juntamente com **find** para pesquisar as concordâncias com o critério dado.[8]

O sistema controla automaticamente as informações referentes a horários de modificação, contagens de links e tamanho de arquivo. Ao contrário, os bits de permissão, propriedade e propriedade de grupo mudam somente quando eles são alterados especificamente por meio dos comandos **chmod**, **chown** e **chgrp**.

chmod: modificando permissões

O comando **chmod** modifica as permissões num arquivo. Somente o proprietário do arquivo e o superusuário podem alterar suas permissões. Para empregarmos o comando em sistemas UNIX antigos, tínhamos que aprender um pouco sobre notação octal, porém as versões atuais aceitam tanto notação octal quanto sintaxe mnemônica. A sintaxe octal geralmente é mais conveniente para administradores, contudo ela somente pode ser usada para especificar um valor absoluto para os bits de permissão. A sintaxe mnemônica pode modificar alguns bits enquanto deixa outros sem modificação.

O primeiro argumento para **chmod** é uma especificação das permissões a serem atribuídas e o segundo argumento e os subseqüentes são os nomes dos arquivos nos quais as permissões devem ser modificadas. No caso octal, o primeiro dígito octal da especificação é para o proprietário, o segundo é para o grupo e o terceiro para o restante. Se quiser ativar os bits setuid, setgid ou sticky, use quatro dígitos octais em vez de três, com os três bits especiais formando o primeiro dígito.

A Tabela 5.4 ilustra as oito possíveis combinações para cada conjunto de três bits, onde `r`, `w` e `x` significam read, write e execute.

Tabela 5.4 Codificação de permissão para o comando **chmod**.

Octal	Binário	Permissões	Octal	Binário	Permissões
0	000	---	4	100	r--
1	001	--x	5	101	r-x
2	010	-w-	6	110	rw-
3	011	-wx	7	111	rwx

Por exemplo, **chmod 711 myprog** dá todas as permissões ao proprietário e permissão somente para execução para o restante.[9]

Os detalhes completos da sintaxe mnemônica **chmod** podem ser encontrados na página de manual referente a **chmod**. Alguns exemplos de especificações mnemônicas são mostrados na Tabela 5.5.

Tabela 5.5 Exemplos de sintaxe mnemônica de **chmod**.

Especificação	Significado
u + w	Acrescenta permissão de gravação para o proprietário do arquivo.
ug=rw,o=r	Dá permissão de leitura/gravação ao proprietário e ao grupo e permissão de leitura aos demais.
a-x	Elimina a permissão de execução para todas as categorias (proprietário/grupo/outros).
ug=srx,o=	Torna o arquivo setuid e setgid e dá permissão de leitura/gravação somente ao proprietário e ao grupo.
g=u	Faz com que as permissões do grupo sejam as mesmas do que as permissões de proprietário.

A parte árdua em relação ao emprego da sintaxe mnemônica é lembrar se **o** significa "owner" (proprietário) ou "other" (outros) ("other" é correto). Lembre-se apenas de **u** e **g** por analogia a UID e GID; somente restará uma possibilidade.

8. Tente usar **find** *ponto_de_montagem* **-xdev -inum** *inode*.
9. Se **myprog** fosse um script de shell, ele precisaria ter ativas tanto a permissão para leitura como para execução. Para que o script seja executado por um interpretador, ele tem de ser aberto e lido como um arquivo de texto. Arquivos binários são executados diretamente pelo kernel e, conseqüentemente, não precisam ter permissão de leitura ativada.

Snow666

Você também pode especificar os modos a serem atribuídos por analogia com um arquivo existente. Por exemplo, **chmod --reference=filea fileb** faz com que o modo de **fileb** seja igual ao de **filea**.

chmod é capaz de atualizar recursivamente as permissões dentro de um diretório com a opção **-R**. Entretanto, isso é mais capcioso do que parece, já que os arquivos e diretórios inclusos talvez não compartilhem os mesmos atributos (por exemplo, alguns poderiam ser arquivos executáveis ao passo que outros são arquivos de texto). A sintaxe mnemônica é particularmente útil com **-R**, pois quaisquer bits cujos valores não ativemos explicitamente são deixados de lado. Por exemplo,

```
% chmod -R g+w mydir
```

acrescenta permissão de gravação para **mydir** e todo o seu conteúdo sem desorganizar os bits de execução de diretórios e programas.

chown: modificando propriedade e grupo

O comando **chown** modifica a propriedade de arquivo e a propriedade de grupo de um arquivo. Sua sintaxe espelha aquela de **chmod**, exceto pelo fato de que o primeiro argumento especifica o novo proprietário e grupo na forma *user.group*. Tanto *user* como *group* podem ser omitidos. Se não houver nenhum grupo, também não é necessário o ponto.

Para modificar o grupo de um arquivo, temos de ser o proprietário do arquivo e pertencer ao grupo que estamos modificando ou então ser o superusuário. Temos de ser o superusuário para modificar a propriedade de um arquivo.

Assim como **chmod**, **chown** oferece o flag recursivo **-R** para modificar as configurações de um diretório de todos os arquivos debaixo dele. Por exemplo, a seqüência

```
# chmod 755 ~matt
# chown -R matt.staff ~matt
```

poderia ser usada para o diretório de usuário de um novo usuário depois de termos copiado os arquivos de inicialização padrão. Certifique-se de não tentar usar **chown** sobre os arquivos ponto do novo usuário através de um comando como

```
# chown -R matt.staff ~matt/.*
```

O padrão concordará com **~matt/..** e, conseqüentemente, acabará modificando as propriedades do diretório-pai e provavelmente os diretórios de origem (home) dos demais usuários.

O UNIX tradicional usa um comando distinto, **chgrp**, para modificar o proprietário do grupo de um arquivo. O Linux também dispõe de **chgrp**. Ele funciona da mesma forma que **chown**; fique à vontade para usá-lo caso ache mais fácil de memorizar.

umask: atribuindo permissões-padrão

Podemos usar o comando de shell embutido chamado **umask** para manipular as permissões-padrão dos arquivos que criamos. **umask** é especificado com um valor octal de três dígitos que representa as permissões a serem retiradas. Quando um arquivo é criado, suas permissões são configuradas para que atendam a tudo que o programa criador solicitar menos aquilo que **umask** proibir. Portanto, os dígitos correspondem às permissões mostradas na Tabela 5.6.

Tabela 5.6 Codificação de permissões do comando **umask**.

Octal	Binário	Permissões	Octal	Binário	Permissões
0	000	rwx	4	100	-wx
1	001	rw-	5	101	-w-
2	010	r-x	6	110	--x
3	011	r--	7	111	---

Por exemplo, **umask 027** autoriza todas as permissões para o proprietário, mas proíbe permissão de gravação para o grupo e não dá nenhum tipo de permissão para os demais. O valor-padrão de **umask** normalmente é 022, que nega permissão de gravação para o grupo e os demais, mas autoriza permissão de leitura.

Snow666

Veja o o Capítulo 6 para obter maiores informações sobre os arquivos de inicialização.

Não há nenhuma maneira de forçarmos os usuários a terem um determinado valor **umask** já que eles sempre podem reinicializá-lo para aquilo que desejarem. Entretanto, podemos fornecer um padrão adequado nos arquivos **.cshr** e **.profile** que damos aos novos usuários.

Flags adicionais

O sistema de arquivos ext2fs do Linux define alguns atributos suplementares que podemos ativar para solicitar semântica especial de sistema de arquivos — onde "solicitar" é a palavra operante já que muitos dos flags não foram, na verdade, implementados. Por exemplo, um flag torna um arquivo somente para anexação e um outro o torna imutável e indelével.

Já que estes flags não se aplicam aos sistemas de arquivos que não são ext2fs, o Linux usa comandos especiais, **lsattr** e **chattr**, para visualizá-los e modificá-los. A Tabela 5.7 apresenta os flags definidos no momento. Informações sobre o estado de implementação de cada flag são esparsas, conflitantes e normalmente desatualizadas, de modo que analisamos o código-fonte para verificar a função de cada flag segundo o kernel versão 2.4.5. Não há nenhuma garantia de que qualquer um dos flags não funcionais será alguma vez implementado.

Tabela 5.7 Flags adicionais ext2fs (segundo o kernel 2.4.5).

Flag	Significado	Funciona?
A	Jamais atualiza o horário de acesso (`st_atime`; para fins de desempenho)	sim
a	Permite gravação somente no modo apêndice (somente root é capaz de configurá-lo)	sim
c	Mantém o conteúdo de arquivo compactado (transparentemente)	não
d	Nenhum backup — faz com que **dump** ignore este arquivo	sim
i	Torna o arquivo imutável e não eliminável (somente root pode ser configurado)	sim
S	Impõe que as modificações sejam gravadas sincronicamente (sem buffering)	mais ou menos
s	Apaga fisicamente (grava por cima) quando eliminado	não
u	Salva o conteúdo na eliminação para permitir a recuperação	não

Com a possível exceção do flag "no backup", não está claro que nenhum desses recursos seja de muita valia no dia-a-dia. Os flags imutáveis e somente anexáveis foram, em grande parte, concebidos como formas de tornar o sistema mais resistente a ataques por parte de hackers ou código hostil. Infelizmente, eles podem confundir o software e somente fornecem proteção contra hackers que não saibam muito sobre o emprego de **chattr -ia**.[10] A experiência prática tem demonstrado que estes flags são mais freqüentemente usados *pelos* hackers do que *contra* os hackers.

A opção **S** para gravações síncronas também merece cautela especial. Já que ela força todas as páginas do sistema de arquivos associadas a um arquivo a serem gravadas imediatamente após modificações, ela poderia parecer oferecer proteção adicional contra perda de dados na eventualidade do sistema dar pau. Entretanto, a ordem das operações para atualizações síncronas é incomum e se tem conhecimento dela confundir o **fsck**; a recuperação de um sistema de arquivos danificado poderia ser, portanto, mais difícil em vez de mais confiável.

5.6 EXERCÍCIOS

E5.1 O que é uma **umask**? Crie uma **umask** que não dê nenhuma permissão ao grupo ou aos demais usuários.

E5.2 Qual a diferença entre hard links e links simbólicos (soft)? Quando é aconselhável usar um tipo ou o outro?

★ **E5.3** Leia a página de manual referente ao arquivo **/etc/fstab**. Crie uma entrada que monte automaticamente uma partição NTFS Windows, **/dev/hda1**, na inicialização. Use o ponto de montagem **/mnt/win_c**.

★ **E5.4** Ao instalar um sistema Linux, é importante criar partições no disco rígido para que haja espaço suficiente para cada sistema de arquivos (**/var**, **/usr** etc.). A distribuição "Foobar Linux" usa os seguintes padrões:

/	100MB
/var	50MB
/boot	10MB
<swap>	128MB
/usr	espaço remanescente

10. O kernel do Linux na verdade suporta um mecanismo de "capacidade" que pode torná-lo mais difícil de desativar esses bits, porém o recurso não está ainda documentado e nem é usado em larga escala.

Snow666

Quais são alguns problemas em potencial com esse arranjo num servidor ocupado?

★ **E5.5** Por que é uma boa idéia colocar algumas partições (como **/var**, **/home** e **swap**) num drive separado dos demais arquivos de dados e programas? O que dizer sobre **/tmp**? Dê razões específicas para cada um dos sistemas de arquivos listados.

★ **E5.6** Crie um script que localize todos os hard links num sistema de arquivos.

★ **E5.7** Apresente comandos para realizar as seguintes tarefas:

a) Configurar as permissões no arquivo **README** para leitura/gravação para o proprietário e de leitura para os demais.

b) Ativar o bit setuid de um arquivo sem modificar (ou conhecer) as permissões atuais.

c) Listar o conteúdo do diretório atual, ordenando-o conforme o horário de modificação e listando por último os modificados mais recentemente.

d) Modificar o grupo de um arquivo chamado **shared** de "user" para "friends".

★ **E5.8** Como você organizaria o software instalado localmente em seu computador Linux caso:

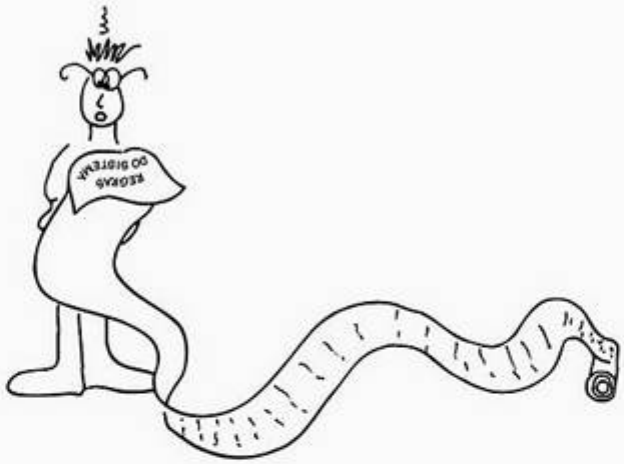
a) Tivesse uma instalação inteiramente em Linux?

b) Tivesse uma instalação com Windows e Linux?

c) Tivesse quatro outras versões do UNIX misturadas em seus computadores Linux?

Snow666

6 Adicionando Novos Usuários



Os tempos dos servidores centralizados com contas de logon para centenas de pessoas já se foram. Apesar disso, adicionar e remover usuários ainda é um conhecimento administrativo essencial. Os administradores precisam ter um entendimento completo do sistema de contas Linux para poder gerenciar serviços de rede e configurar contas de maneira apropriada para o ambiente computacional local.

Saneamento de contas também é um determinante fundamental da segurança do sistema. Contas usadas com pouca freqüência são os alvos principais para os hackers, assim como as contas com senhas facilmente decifráveis. Mesmo que você use ferramentas automatizadas do sistema para acrescentar e eliminar usuários, é importante compreender as mudanças subjacentes que essas ferramentas estão realizando.

Neste capítulo, examinaremos o modelo subjacente que as ferramentas automatizadas implementam, depois descreveremos as ferramentas em si (**useradd**, **userdel** etc.). A ferramenta-padrão **useradd** é, na verdade, relativamente boa e deve ser suficiente para atender às necessidades da maioria das instalações. Infelizmente, **userdel** não é tão completa como gostaríamos que fosse.

6.1 O Arquivo /etc/passwd

O arquivo **/etc/passwd** representa uma lista de usuários reconhecidos pelo sistema. O sistema consulta o arquivo no momento do login para determinar o UID do usuário e para verificar a sua senha. Cada linha no arquivo indica um usuário e contém sete campos separados por sinais de dois-pontos (:):

- Nome de login
- Senha criptografada (a menos que um arquivo de senha-sombra seja usado)
- Número UID
- Número GID padrão
- Informações "GECOS": nome completo, empresa, ramal, telefone residencial
- Diretório inicial
- Shell de login

Por exemplo, as linhas a seguir são todas entradas **/etc/passwd** válidas:

```
root:lga5FjuGpZ2so:0:0:The System,,x6096,:/:/bin/csh
jl:x:100:0:Jim Lane,ECT8-3,,:/staff/jl:/bin/sh
dotty:$1$Ce8QpAQI$L.DvJEWittlWetKTMLXFZO/:101:20::/home/dotty:bin/csh
```

Snow666

À medida que o hardware dos computadores se tornou mais rápido, ficou cada vez mais perigoso deixar senhas criptografadas às claras. O Linux permite que você oculte as senhas criptografadas colocando-as num arquivo separado não legível por todo o mundo. Isso é conhecido como um mecanismo de senhas-sombra e é o padrão na maioria das distribuições.

O sistema de senhas-sombra faz mais sentido quando explicado como uma extensão do tradicional **/etc/passwd** (como ele era historicamente), de modo que iremos protelar a discussão sobre este recurso até a seção 6.2. Uma discussão mais genérica sobre implicações de segurança das senhas-sombra pode ser encontrada no Capítulo 21.

O conteúdo de **/etc/passwd** é normalmente compartilhado entre vários sistemas com um banco de dados como NIS ou LDAP. Veja o Capítulo 18, *Compartilhando Arquivos de Sistema*, para obter maiores informações.

As seções a seguir discutem os campos de **/etc/passwd** de maneira mais detalhada.

Nome de login

Os nomes de login (também conhecidos como nomes de usuário) têm de ser exclusivos e não superiores a 32 caracteres de comprimento. Eles podem conter quaisquer caracteres exceto dois-pontos e linhas novas.

Algumas versões mais antigas do UNIX limitam os caracteres permissíveis a caracteres alfanuméricos e impõem um limite de comprimento de 8 caracteres. Se você usa NIS, os nomes de login são limitados a 8 caracteres independentemente do sistema operacional.

Sugerimos enfaticamente que você obedeça aos limites mais restritivos mesmo no Linux. Tal política evitará possíveis conflitos com software mais antigo e garantirá que os usuários possam ter o mesmo nome de login em cada uma das máquinas. Lembre-se, o fato de termos um ambiente homogêneo hoje não significa que isso será verdade no futuro.

Os nomes de login diferenciam letras maiúsculas de minúsculas; entretanto, a maioria dos sistemas de correio (inclusive **sendmail**) não dá atenção para isso. Não sabemos de nenhum caso de problemas causados por mistura de caixas em nomes de login, porém os nomes minúsculos são tradicionais e mais fáceis de serem digitados.

Nomes de login devem ser fáceis de serem memorizados, de modo que seqüências aleatórias de letras não criam bons nomes de login. Sugerimos que você evite "cognomes" e apelidos bonitinhos, mesmo que sua empresa seja relativamente informal. Eles não são tão divertidos assim e tendem a provocar gozações; nomes como DarkLord e QTPie aparecem na frente de @aol.com. Mesmo que os seus usuários não tenham auto-estima, pense pelo menos na credibilidade geral de sua instalação.

Já que os nomes de login são normalmente usados como endereços de e-mail, é útil estabelecer uma maneira-padrão de formá-los. Deveria ser possível para os usuários fazer suposições baseadas em fatos. Nomes, sobrenomes, iniciais ou alguma combinação destes são esquemas de nomes razoáveis.

Verifique o Capítulo 19 para obter maiores informações sobre nomes alternativos de correio.

Qualquer esquema fixo para seleção de nomes de logins resulta eventualmente em nomes duplicados ou nomes que são muito longos, de modo que às vezes você terá de abrir exceções. No caso de um nome longo, podemos usar o arquivo **/etc/mail/aliases** para equiparar duas versões do nome, pelo menos no que se refere a correio.

Suponha, por exemplo, que você use a primeira letra do nome e o sobrenome de um funcionário como paradigma. Brent Browning seria, portanto, "bbrowning", que tem 9 caracteres e, portanto, é muito longo. Em vez disso, poderíamos atribuir ao usuário o login "brentb", deixando "bbrowning" como uma entrada do arquivo **aliases**:

```
bbrowning: brentb
```

Se sua instalação tiver um arquivo alias de correio global, cada novo nome de login deve ser distinto de qualquer alias neste arquivo. Caso contrário, a mensagem será enviada ao nome alternativo, e não ao novo usuário.

É comum para grandes instalações implementar um esquema de endereços de e-mail com nomes completos (por exemplo, John.Q.Public@mysite.com) que oculta nomes de login do mundo externo. Essa é uma boa idéia, porém ela realmente não elimina qualquer dos conselhos sobre atribuição de nomes dados anteriormente. Se por nenhuma outra razão a não ser a sanidade dos administradores, é melhor que os nomes de login tenham uma correspondência clara e previsível com os verdadeiros nomes dos usuários.

Se você tiver mais do que uma máquina, os nomes de login devem ser exclusivos em dois sentidos. No primeiro deles, um usuário deve ter o mesmo nome de login em todas as máquinas. Essa regra é seguida mais por conveniência, tanto para você como para o usuário.

Snow666

Verifique a seção 21.7, no Capítulo 21, para uma discussão sobre questões de equivalência de logins.

Em segundo lugar, um determinado nome de login deve sempre se referir à mesma pessoa. Alguns comandos (como **ssh**) podem ser configurados para validar usuários remotos tomando com base seus nomes de login. Mesmo que scott@boulder e scott@refuge fossem duas pessoas diferentes, uma poderia ser capaz de entrar na conta do outro sem fornecer uma senha caso os sistemas não fossem configurados apropriadamente.

A experiência também mostra que nomes duplicados podem induzir a confusão de e-mails. O sistema de e-mail poderia perfeitamente esclarecer de qual scott estamos tratando, porém os usuários normalmente enviariam mensagens para o endereço errado.

Senha criptografada

Um rápido lembrete antes de entrarmos de cabeça nos detalhes das senhas: a maioria dos sistemas na verdade mantém senhas criptografadas em **/etc/shadow**, e não em **/etc/passwd**. Entretanto, os comentários desta seção se aplicam independentemente de onde as senhas sejam efetivamente mantidas.

/etc/passwd armazena senhas numa forma criptografada. A menos que você consiga executar algoritmos de criptografia em sua cabeça (queremos conhecê-lo!), terá de configurar o conteúdo deste campo usando o comando **passwd** (**yppasswd** caso use NIS) ou copiando uma string de senha criptografada de uma outra conta.

Se você editar **/etc/passwd** para criar uma conta, coloque um asterisco (*) no campo de senha criptografada. O asterisco impede uso desautorizado da conta até que você tenha configurado uma senha real. Jamais deixe esse campo vazio — isto gera uma brecha enorme em termos de segurança, pois não é necessária nenhuma senha para acessar a conta.[1]

A maioria das distribuições Linux (inclusive SuSE e Debian) usa como padrão criptografia DES para senhas. Esse padrão limita o comprimento de senhas não criptografadas a 8 caracteres. Senhas mais longas são aceitas, porém somente os 8 primeiros caracteres têm significação. Os caracteres adicionais são silenciosamente ignorados.

Um sistema de senhas alternativo baseado no algoritmo de hashing seguro MD5 é o padrão no Red Hat e uma opção na maioria das demais versões do Linux. O MD5 não é criptograficamente "melhor" do que o DES, porém o esquema MD5 permite senhas de comprimento arbitrário. Senhas mais longas são mais seguras — se você realmente as usar. Já que o emprego de MD5 não fará mal e talvez ajude, o recomendamos para todas as instalações.

Todas as principais distribuições Linux reconhecerão senhas MD5 no arquivo **passwd** caso elas estejam presentes. Não é necessário que todas as senhas do sistema usem a mesma forma de criptografia.

Para convencer SuSE e Debian a gerar senhas MD5, bem como reconhecê-las, você tem de rever *todos* os arquivos em **/etc/pam.d**. Acrescente a palavra-chave `md5` para finalizar as linhas que começam com `password` e mencione `pam_unix.so` ou `pam_pwcheck.so`. Por exemplo, alteraríamos a linha Debian

```
password required pam_unix.so nullok obscure min=4 max=8
```

para

```
password required pam_unix.so nullok obscure min=4 max=8 md5
```

O argumento `max=8` parece, à primeira vista, limitar o comprimento da senha a 8 caracteres. Entretanto, isso não se encontra documentado e parece não ter nenhum efeito sobre senhas MD5 ativas. Ignore-o. No Debian, podemos atingir o mesmo efeito de habilitar senhas MD5, utilizando o utilitário de administração de pacotes **dpkg-reconfigure**:

```
# dpkg-reconfigure passwd
```

Esse comando executará um diálogo em que o administrador poderá escolher habilitar senhas MD5 e também senhas-sombra (**/etc/shadow**).

No Red Hat, o comando **authconfig** aceita o argumento **--enablemd5** para ativar a geração de senhas MD5.[2] Acrescente **--kickstart** para impedir que a irritante interface interativa apareça.

Senhas criptografadas no arquivo **/etc/passwd** normalmente têm comprimento constante (34 caracteres para o MD5, 13 para o DES) independentemente do comprimento da senha não criptografada. As senhas são criptogra-

1. Mesmo que você esteja usando senhas-sombra, é prudente ser um tanto cauteloso em relação à saúde do arquivo **/etc/passwd**. Jamais saberemos quando algum programa ou script obsoleto irá espioná-lo para tomar alguma decisão em termos de segurança.
2. Há também uma opção **--disablemd5**, embora não consigamos imaginar uma razão para usá-la.

www.snow666.blogspot.com
Snow666

fadas em combinação com um "tempero" aleatório para que uma dada senha possa corresponda a várias formas criptografadas distintas (normalmente 64). Se por acaso dois usuários selecionarem a mesma senha, este fato normalmente não poderá ser descoberto por inspeção do arquivo **passwd**. As senhas MD5 são fáceis de serem localizadas, pois elas sempre se iniciam com um 1.

Número UID

Os UIDs são inteiros de 32 bits não sinalizados que podem representar os valores de 0 a 4.294.967.296. Entretanto, devido a questões de interoperabilidade com sistemas mais antigos, sugerimos limitar, se possível, o UID máximo de sua instalação a 32.767.

Por definição, o usuário root tem um UID 0. A maioria dos sistemas também define pseudo-usuários bin, daemon e talvez alguns outros mais. É costumeiro colocar tais logins fictícios no início do arquivo **/etc/passwd** e dar a eles UIDs baixos. Para ter bastante espaço para quaisquer usuários não humanos que você queira adicionar no futuro, recomendamos que atribua UIDs aos usuários reais iniciando em 100 (ou maior).

*Veja o Capítulo 3 para obter maiores informações sobre **sudo**.*

Jamais será uma boa idéia ter várias contas com UID 0. Embora possa parecer conveniente ter vários logins root com diferentes shells e/ou senhas, esta configuração apenas cria mais brechas de segurança em potencial e faz com que você tenha de assegurar vários logins. Se as pessoas precisam ter maneiras alternativas para logar como root, é melhor que elas usem um programa como o **sudo**.

Evite reaproveitar UIDs pelo maior tempo possível, mesmo que sejam UIDs de pessoas que tenham saído da organização e tenham suas contas eliminadas permanentemente. Essa precaução impede confusão caso arquivos sejam restaurados posteriormente de backups, em que os usuários sejam identificados pelo UID em vez de um nome de login.

Veja o Capítulo 17 para obter maiores informações sobre NFS.

Os UIDs devem ser mantidos exclusivos ao longo de toda a sua organização. Ou seja, um determinado UID deve referir-se ao mesmo nome de login e à mesma pessoa em cada máquina. O ato de não manter UIDs distintos pode resultar em problemas de segurança com sistemas como o NFS e também pode criar confusão quando um usuário se transferir de um grupo a outro.

Pode ser difícil manter UIDs exclusivos quando grupos de máquinas são administrados por pessoas diferentes da empresa. Os problemas podem ser tanto técnicos como políticos. A melhor solução é ter um banco de dados centralizado contendo um registro para cada usuário e reforçar a exclusividade. (Usamos um banco de dados caseiro para esse problema.) Um esquema mais simples é atribuir a cada grupo dentro de uma empresa um intervalo de UIDs e deixar que cada grupo gerencie o seu próprio conjunto. Essa solução mantém os espaços de UID separados (uma exigência caso vá usar NFS para compartilhar sistemas de arquivos), porém não atende à questão paralela de nomes de login exclusivos.

Número GID Padrão

Como um UID, um número ID de grupo é um inteiro de 32 bits não sinalizado. O GID 0 é reservado para o grupo chamado "root". GID 1 é o grupo "bin" e GID 2 é o grupo "daemon".

Veja O Capítulo 5 para obter maiores informações sobre diretórios setgid.

Os grupos são definidos em **/etc/group**, com o campo GID em **/etc/passwd** fornecendo o GID padrão (ou "efetivo") no momento do login. O GID padrão não é tratado especialmente quando o acesso é determinado;[3] ele é relevante somente para a criação de arquivos e diretórios novos. Arquivos novos normalmente são de propriedade do grupo efetivo do usuário. Porém, em diretórios nos quais o bit setgid (02000) tiver sido ativado e em sistemas de arquivos montados com a opção **grpid**, arquivos novos terão como padrão o grupo de seus diretórios-pai.

Campo GECOS

O campo GECOS é comumente usado para registrar informações pessoais sobre cada usuário. Ele não possui uma sintaxe bem definida. O campo GECOS originalmente armazenava as informações de login necessárias para transferir tarefas em segundo plano de sistemas UNIX na Bell Labs para um mainframe executando o sistema operacional GECOS (General Electric Comprehensive Operating System); hoje, somente o nome permanece.

Embora possamos usar quaisquer convenções de formatação que queiramos, **finger** interpreta entradas GECOS separadas por vírgulas na seguinte ordem:

---

3. O Linux permite a um usuário estar em até 32 grupos simultaneamente, e todos os grupos são considerados quando os cálculos de acesso são realizados.

Snow666

- Nome completo (normalmente o único campo usado)
- Prédio e número do escritório
- Ramal do telefone comercial
- Telefone residencial

O comando **chfn** permite aos usuários modificar suas próprias informações GECOS. **chfn** é útil para manter atualizados itens como números de telefones, porém ele pode ser usado de maneira abusiva: um usuário pode alterar as informações para que tornem-se obscenas ou incorretas. Nosso centro de computação acadêmico, que atende a hordas de alunos, desabilitou o comando **chfn**.

Diretório inicial

Os shells dos usuários mudam para seus diretórios iniciais via **cd** quando é feita a entrada no sistema. Se o diretório inicial de um usuário estiver faltando no momento do login, o sistema imprimirá uma mensagem como "no home directory".[4] Os sistemas Linux geralmente permitem que o login prossiga e colocam o usuário no diretório-raiz. Para desabilitar logins que não possuam um diretório inicial válido, configure `DEFAULT_HOME` para `no` em **/etc/login.defs**.

Esteja ciente de que se os diretórios são montados através de um sistema de arquivos de rede, eles podem estar indisponíveis no caso de problemas de rede ou servidor.

Shell de login

O shell de login normalmente é um interpretador de comandos como o shell Bourne ou o shell C (**/bin/sh** ou **/bin/csh**), porém pode ser qualquer programa. **bash** é o padrão e é utilizado se **/etc/passwd** não especifica um shell de login. Em sistemas Linux, **csh** e **sh** são realmente apenas links para **tcsh** (um superconjunto do shell C) e **bash** (o shell GNU "Bourne novamente"), respectivamente. A maioria das distribuições também fornece uma versão de domínio público do shell Korn, **ksh**.

*Recomendamos **tcsh** como shell padrão para novos usuários.*

Os usuários podem alterar seus shells com o comando **chsh**. O arquivo **/etc/shells** contém uma lista de shells "válidos" que **chsh** permitirá que os usuários selecionem; SuSE e Debian impõem esta lista, porém o Red Hat apenas o alerta caso o shell selecionado não se encontre na lista. Se você acrescentar entradas ao arquivo **shells**, certifique-se de usar caminhos absolutos já que **chsh** e outros programas têm a expectativa de usá-lo.

6.2 O ARQUIVO /ETC/SHADOW

O arquivo **/etc/shadow** é legível somente pelo superusuário e serve para manter senhas criptografadas seguras quanto a acesso não autorizado. Ele também fornece informações de contas que não se encontra disponível em **/etc/passwd**. O emprego de senhas-sombra é padrão em algumas distribuições e configurado como pacote opcional em outros. Mesmo quando elas são opcionais, é uma boa idéia tratá-las como se fossem padrão.

Quando as senhas-sombra estão em uso, os campos de senha com o estilo antigo contidos em **/etc/passwd** devem sempre conter um `x`.

O arquivo **shadow** não é um superconjunto do arquivo **passwd**, e o arquivo **passwd** não é gerado dele; você tem de manter ambos os arquivos (ou usar ferramentas como **useradd** que mantêm ambos em seu nome). Assim como **/etc/passwd**, **/etc/shadow** contém uma linha para cada usuário. Cada linha contém campos, separados por dois-pontos (:) :

- Nome de login
- Senha criptografada
- Data da última mudança de senha
- Número mínimo de dias entre mudanças de senha
- Número máximo de dias entre mudanças de senha
- Número de dias antecipados para alertar os usuários sobre a expiração de suas senhas
- Número de dias após a expiração da senha que a conta será desabilitada

4. Esta mensagem aparece quando logamos na console ou num terminal, porém não aparece ao logarmos por meio de um gerenciador de exibição como **xdm**, **gdm** ou **kdm**. Você não somente não verá a mensagem, como também em geral será colocado fora do sistema imediatamente devido à incapacidade do gerenciador de exibição gravar no diretório apropriado (por exemplo, **~/.gnome**).

Snow666

- Data de expiração da conta
- Um campo reservado que se encontra atualmente vazio

*Veja a página 74 para obter maiores informações sobre **usermod**.*

Os únicos campos em que se exige que não estejam vazios são os referentes a nome de usuário e senha. Campos de datas absolutos em **/etc/shadow** são especificados em termos de dias (*não* em segundos) desde 1.º de janeiro de 1970, que não é uma maneira-padrão de reconhecimento de horários em sistemas UNIX. Felizmente, podemos usar o programa **usermod** para configurar o campo de expiração.

Uma entrada **shadow** típica seria parecida com o seguinte:

```
millert:inNO.VRsclWn.:11508:0:180:14::12417:
```

A seguir uma descrição mais completa de cada campo:

- O nome de login é o mesmo que em **/etc/passwd**. Esse campo simplesmente conecta as entradas de usuário em **passwd** e **shadow**.
- A senha criptografada é idêntica em conceito e execução àquela previamente armazenada em **/etc/passwd**.
- O campo da última mudança indica o momento no qual a senha do usuário foi modificada pela última vez. Esse campo é geralmente preenchido por **/usr/bin/passwd**.
- O quarto campo configura o número de dias que devem decorrer entre mudanças de senha. A partir do momento em que os usuários mudam suas senhas, eles não poderão alterá-las antes que um período especificado tenha passado. Esse recurso parece inútil, e acreditamos que ele poderá ser um tanto perigoso quando uma intrusão na segurança do sistema tiver ocorrido. Recomendamos configurar esse campo em 0.
- O quinto campo configura o número máximo de dias permitido entre mudanças de senha. Esse recurso permite ao administrador forçar a expiração de senhas; consulte a página 491 para obter maiores informações. O verdadeiro número máximo de dias para expiração é a soma desse campo e do sétimo campo (período de tolerância).
- O sexto campo configura o número de dias antes da expiração da senha que o programa **login** deve começar a alertar o usuário da expiração iminente.
- O sétimo campo especifica o número máximo de dias que se pode esperar após o limite de existência de uma senha ter sido atingido antes de tratar o login como expirado. Isso parece ser um recurso sem sentido, e a documentação do Linux é extremamente vaga em relação ao seu propósito. Tivemos de entrar no código-fonte para descobrir o que ele fazia.
- O oitavo campo especifica o dia (a partir de 1.º de janeiro de 1970) no qual uma conta de usuário irá expirar. Talvez o usuário não consiga logar após essa data até que o campo tenha sido reinicializado por um administrador. Se o campo for deixado em branco, a conta jamais expirará.
- O nono campo é reservado para uso futuro.

Agora que sabemos o significado de cada um desses campos, vejamos nosso exemplo novamente:

```
millert:inNO.VRsclWn.:11508:0:180:14::12417:
```

Nesse caso, o usuário millert alterou sua senha pela última vez em 4 de julho de 2001. A senha tem de ser modificada novamente dentro de 180 dias, e millert receberá alertas de que a senha precisa ser modificada durante as duas últimas semanas desse período. A conta expira em 31 de dezembro de 2003.

Podemos usar o utilitário **pwconv** para reconciliar o conteúdo do arquivo **shadow** com aquele do arquivo **passwd**, selecionando quaisquer novas adições e eliminando usuários que não se encontrem mais listados em **passwd**. **pwconv** preenche a maioria dos parâmetros shadow por meio dos padrões definidos em **/etc/login.defs**.

6.3 O Arquivo /etc/group

O arquivo **/etc/group** contém os nomes dos grupos UNIX e uma lista dos membros de cada grupo. Por exemplo:

```
wheel:*:10:root,evi,garth,trent
csstaff:*:100:lloyd,evi
student:*:200:dotty
```

Cada linha representa um grupo e contém quatro campos:

www.snow666.blogspot.com
Snow666

- Nome do grupo
- Senha criptografada (vestigial e raramente usada)
- Número GID
- Lista de membros, separados por vírgulas (atenção para não adicionar espaços)

Assim como acontece em **/etc/passwd**, os campos são separados por dois-pontos. Os nomes de grupos devem ser limitados a oito caracteres para fins de compatibilidade, embora o Linux não exija isto. Embora seja possível introduzir uma senha de grupo (para permitir a usuários não pertencentes a um grupo mudarem para ele por meio do comando **newgrp**), isto raramente é feito.[5] A maioria das instalações coloca asteriscos no campo de senhas (Debian, RedHat e SuSE colocam o caractere 'x' no campo de senhas), porém é seguro deixar o campo em branco caso queiramos. O comando **newgrp** não mudará para um grupo a menos que o usuário já se encontre listado como membro desse grupo.

Como acontece com nomes de usuários e UIDs, os nomes de grupos e GIDs devem ser mantidos consistentes entre as máquinas que compartilham arquivos por meio de um sistema de arquivos. Pode ser difícil manter a consistência num ambiente heterogêneo, já que vários sistemas operacionais usam GIDs diferentes para os mesmos nomes de grupos. Descobrimos que a melhor maneira de lidar com essa questão é evitar usar um grupo de sistema como grupo de login padrão para um usuário.

Se um usuário tem como padrão um determinado grupo em **/etc/passwd**, mas não parece estar nesse grupo de acordo com **/etc/group**, **/etc/passwd** fica com o argumento. Associações a grupos concedidas no momento do login é realmente a união daqueles encontrados nos arquivos **passwd** e **group**. Entretanto, é uma boa idéia manter os dois arquivos consistentes.

Para minimizar o potencial de possíveis conflitos com GIDs fornecidos por terceiros, sugerimos iniciar grupos locais em GID 100.

A tradição do UNIX sempre foi a de acrescentar novos usuários a um grupo que representa sua categoria genérica como "estudantes" ou "financeiro". Entretanto, vale a pena notar que tal convenção aumenta a probabilidade de os usuários serem capazes de lerem os arquivos uns dos outros devido à configuração descuidada de permissões, mesmo que esta não seja realmente a intenção do proprietário. Para evitar esse problema, recomendamos que você crie um grupo exclusivo para cada usuário. Você pode usar o mesmo nome tanto para o usuário como para o grupo.

O grupo pessoal de um usuário deveria conter somente este usuário. Se quiser permitir aos usuários compartilharem arquivos pelo mecanismo de grupos, crie grupos separados para esta finalidade. A idéia implícita nos grupos pessoais não é a de desencorajar o uso de grupos em si — é a de simplesmente estabelecer um grupo-*padrão* mais restritivo para cada usuário para que estes arquivos não sejam compartilhados inadvertidamente.

A versão Red Hat de **useradd** tem como padrão a criação de grupos pessoais para novos usuários. Embora os utilitários de manutenção de usuário do Linux sejam muito similares entre as diversas distribuições, isto parece ser uma característica específica do Red Hat.

No Debian, o utilitário **adduser** pode ser configurado, por meio do arquivo **/etc/adduser.conf**, para criar um grupo pessoal para cada novo usuário adicionado ao sistema. A linha de configuração no arquivo **/etc/adduser.conf** é:

```
USERGROUPS=yes
```

6.4 Adicionando Usuários

Antes de criar uma conta para um novo usuário numa instalação corporativa, governamental ou educacional, é muito importante que o usuário assine e date uma cópia do acordo de usuário local e políticas a serem observadas. (O quê?! Você não tem esses documentos? Verifique o Capítulo 29 para obter maiores informações sobre o motivo para você precisar deles e o que colocar neles.)

Os usuários não têm nenhuma razão em particular para querer assinar um acordo de políticas gerais; portanto, é para seu benefício garantir que eles assinem esses documentos enquanto você ainda tem um poder de alavancagem. Acreditamos que dá mais trabalho garantir um acordo assinado após uma conta ter sido liberada. Se seu processo permitir, é melhor ter a papelada pronta e assinada antes da criação da conta.

5. A única razão que temos notícia para alguém querer usar o comando **newgrp** no Linux seria para configurar o grupo padrão de arquivos recém-criados.

www.snow666.blogspot.com
Snow666

Mecanicamente, o processo de acrescentar um novo usuário consiste em três etapas exigidas pelo sistema, duas que estabelecem um ambiente útil para o novo usuário e várias outras etapas adicionais para sua própria conveniência como administrador.

É necessário:

- Editar os arquivos **passwd** e **shadow** para definir a conta do usuário.
- Configurar uma senha inicial.
- Criar e aplicar os comandos **chown** e **chmod** ao diretório inicial do usuário.

Para o usuário:

- Copiar arquivos de inicialização padrão para o diretório inicial do usuário.
- Configurar o diretório inicial de correio do usuário e estabelecer nomes alternativos para correio.

Para você:

- Adicionar o usuário ao arquivo **/etc/group**.
- Configurar quotas de disco.
- Verificar se a conta está configurada corretamente.
- Adicionar as informações de contato de usuário e estado da conta ao seu banco de dados.

O comando **useradd** e seus irmãos são capazes de executar algumas dessas etapas para você, mas nas próximas seções percorreremos as etapas como se você as tivesse executando manualmente. Isso é feito mais para você poder entender o que as ferramentas estão fazendo. Na vida real, em geral é preferível (mais rápido e menos sujeito a erros) executar **useradd** ou algum script caseiro semelhante.

Você tem de executar cada uma das etapas como root ou usar um programa como o **sudo** que lhe permita executar comandos como root.

Editando os arquivos passwd e shadow

Para editar de maneira segura o arquivo **passwd**, execute **vipw** para chamar um editor de texto numa cópia dele (arquivo **passwd**). O editor-padrão é o **vi**, porém podemos especificar um editor diferente configurando o valor de sua variável de ambiente EDITOR. A existência do arquivo de edição temporário serve como um bloqueio; **vipw** permite que somente uma pessoa por vez edite o arquivo **passwd** e ele impede que os usuários modifiquem suas senhas enquanto o arquivo **passwd** estiver sendo usado. Quando o editor finalizar o processo, **vipw** substituirá o arquivo **passwd** original pela sua cópia editada.

Por exemplo, adicionar a linha a seguir a **/etc/passwd** definiria uma conta chamada "tyler":

```
tyler:*:103:100:Tyler Stevens, ECEE 3-27, x7919,:/home/staff/tyler:/bin/tcsh
```

Também adicionamos a **/etc/shadow** uma entrada correspondente, executando **vipw -s**:

```
tyler:*::::::12417:
```

Essa linha **shadow** para "tyler" não possui senha criptografada nem data de expiração de senha e configura a expiração da conta para 31 de dezembro de 2003.

Configurando uma senha inicial

O root pode modificar a senha de qualquer usuário pelo comando **passwd**:

```
# passwd user
```

Regras para selecionar senhas adequadas são apresentadas na seção 21.4, Capítulo 21.

passwd lhe solicita que você introduza uma nova senha e que a digite novamente para fins de confirmação. Se você optar por uma senha curta, toda em letras minúsculas ou de alguma outra forma obviamente inadequada, **passwd** reclamará e lhe solicitará que use algo mais complexo. O Red Hat e o SuSE verificam senhas esperadas comparando-as com aquelas contidas num dicionário para aumentar a segurança.

O utilitário **mkpasswd** que vem junto com o pacote **expect** de Don Libes torna mais fácil a geração de senhas aleatórias para novos usuários. Para melhor ou para pior, a atribuição de uma senha aleatória "força" os usuários novos a alterar suas senhas imediatamente, já que as senhas aleatórias são difíceis de serem memorizadas. Não confun-

Snow666

da **mkpasswd** do **expect** com o comando **mkpasswd** padrão que simplesmente codifica uma determinada string como senha.

Jamais deixe uma conta nova — ou qualquer conta que tenha acesso a um shell — sem uma senha.

Criando o diretório inicial do usuário

Qualquer diretório que criamos como root pertence inicialmente à root, de modo que temos de modificar o seu proprietário e grupo por meio do comando **chown**. A seqüência de comandos a seguir cria um diretório inicial apropriado para nosso usuário de exemplo:

```
# mkdir /home/staff/tyler
# chown tyler.staff /home/staff/tyler
# chmod 700 /home/staff/tyler
```

Copiando arquivos de inicialização padrão

Podemos personalizar alguns comandos e utilitários colocando arquivos de configuração no diretório inicial de um usuário. Os arquivos de inicialização começam com um ponto e finalizam com as letras **rc**, abreviação de "run command", uma relíquia do sistema operacional CTSS. O ponto inicial faz com que **ls** suprima esses arquivos das listagens do diretório a menos que seja usada a opção **-a**; os arquivos são considerados "desinteressantes". A Tabela 6.1 lista alguns arquivos de inicialização comuns.

Tabela 6.1 Arquivos de inicialização comuns e suas aplicações.

Comando	Nome de arquivo	Empregos típicos
csh/tcsh	**.login**	Configura o tipo de terminal (se necessário).
		Configura as variáveis de ambiente.
		Configura as opções do **biff** e **mesg**.
	.cshrc	Configura nomes de comandos alternativos.
		Configura o caminho de busca.
		Configura o valor **umask** para controlar permissões.
		Configura `cdpath` para buscas de nomes de diretórios.
		Configura o prompt, histórico e variáveis para salvar.
	.logout	Configura os lembretes "a fazer".
		Limpa a tela.
sh	**.profile**	Similar a **.login** e **.cshrc** para **sh**.
bash[a]	**.bashrc**	Similar a **.cshrc** para **bash**.
	.bash_profile	Similar a **.login** para **bash**.
vi	**.exrc**	Configura as opções do editor **vi**.
emacs	**.emacs**	Configura as opções do editor **emacs**.
		Configura as vinculações de teclas de **emacs**.
mail/mailx	**.mailrc**	Define nomes de e-mail pessoais alternativos.
		Configura as opções do leitor de mensagens.
xrdb	**.Xdefaults**	Especifica configuração X11: fontes, cores, etc.
startx	**.xinitrc**	Especifica o ambiente X11 inicial.
	.Xclients	Especifica o ambiente X11 inicial (Red Hat).
xdm	**.xsession**	Especifica o ambiente X11 inicial.

a. **bash** também lerá **.profile** ou **/etc/profile** na emulação de **sh**.

Caso você já não tenha um bom conjunto de arquivos de inicialização padrão, **/usr/local/lib/skel** é um bom lugar para colocá-los. Copie alguns arquivos para usar como ponto de partida e modifique-os com um editor de texto. Talvez você queira começar com arquivos oferecidos pelo fornecedor no diretório **/etc/skel**. Certifique-se de configurar um valor razoável para o **umask** de um novo usuário (sugerimos 077, 027 ou 022, dependendo da facilidade e tamanho de sua instalação).

www.snow666.blogspot.com
Snow666

Dependendo do shell do usuário, **/etc** poderá conter arquivos de inicialização válidos para todo o sistema que são processados antes dos arquivos de inicialização do usuário. Por exemplo, **sh** (**bash**, na verdade) lê **/etc/profile** antes de processar **~/.profile**. Para outros shells, consulte a página de manual correspondente ao shell em questão para obter maiores detalhes.

A seqüência de comandos para instalar arquivos de inicialização para o novo usuário tyler ficaria parecida com o seguinte:

```
# cp /usr/local/lib/skel/.[a-zA-Z]* ~tyler
# chmod 644 ~tyler/.[a-zA-Z]*
# chown tyler ~tyler/.[a-zA-Z]*
# chgrp staff ~tyler/.[a-zA-Z]*
```

Note que não podemos usar

```
# chown tyler ~tyler/.*
```

pois tyler ficaria proprietário não somente de seus próprios arquivos como também do diretório-pai ".." (**/home/staff**). Isso é um erro muito comum e perigoso em termos de administração de sistemas.

Selecionando o diretório inicial de e-mail do usuário

É conveniente para cada usuário receber e-mails somente em uma máquina. Esse esquema é normalmente implementado por meio de uma entrada no arquivo de nomes alternativos global **/etc/mail/aliases** ou **sendmail** `userDB` no servidor de e-mail central. Veja o Capítulo 19 para obter informações gerais sobre e-mail; as diversas maneiras de implementar diretórios iniciais para e-mail são discutidas a partir da seção 19.3.

Editando o arquivo /etc/group

Para continuarmos o processamento do novo usuário tyler, devemos adicionar o seu nome de login à lista de usuários do grupo 100, já que este era o grupo-padrão ao qual o atribuímos no arquivo **/etc/passwd**. Na realidade, tyler estará no grupo 100 independentemente de ele estar listado ou não em **/etc/group**, pois sua entrada **passwd** já atribuiu a ele esta associação. Entretanto, essa informação deve ser introduzida em **/etc/group** para que sempre saibamos exatamente quais usuários pertencem a quais grupos. [6]

Configurando cotas de disco

Se sua instalação usa cotas de disco, você deve configurar limites de cotas para cada conta nova pelo comando **edquota**. **edquota** pode ser usado interativamente, porém é mais comumente usado no modo "protótipo" para modelar as cotas do novo usuário após aquelas de algum outro. Por exemplo, o comando

```
# edquota -p proto-user new-user
```

configura as quotas do novo usuário (*new-user*) para que sejam iguais a de *proto-user*. Essa maneira de aplicação de **edquota** é especialmente útil em scripts.

Já que o espaço em disco é barato nos dias de hoje, não somos muito partidários ao emprego de cotas de disco. Elas normalmente parecem trazer mais problemas do que soluções e impõem ao administrador uma carga adicional em termos de suporte. Retornando ao tempo em que usávamos cotas (muitos anos atrás), nós fizemos a manutenção de diversas contas que existiam apenas para servir como protótipos de cotas de usuário.

Verificando o novo login

Para verificar se uma nova conta foi configurada apropriadamente, primeiramente saia do sistema, depois entre novamente como o novo usuário e execute os seguintes comandos:

```
% pwd          /* Para verificar o diretório de usuário */

% ls -la       /* Verifica o proprietário/grupo de arquivos dos arquivos de inicialização */
```

Você precisará notificar os novos usuários a respeito de seus nomes de logins e senhas iniciais. Muitas instalações enviam essas informações via e-mail porém, por questões de segurança, isso nem sempre é uma boa idéia. A conta de um novo usuário pode ser comprometida e invadida (backdoors) antes mesmo de ele conseguir fazer o lo-

6. O kernel na verdade não se preocupa com o que se encontra em **/etc/passwd** ou **/etc/group**; ele somente se importa com números puros UID e GID. **passwd** e **group** armazenam informações de contas para ser empregadas por software de alto nível como **login**. Veja a página 85 para obter maiores detalhes sobre o processo de login.

Snow666

gin. Esse também é um bom momento de direcionar os usuários para documentação adicional sobre costumes locais caso ela realmente exista.

Verifique o Capítulo 19 para obter maiores informações sobre contratos de usuários por escrito.

Se sua instalação exigir que os usuários assinem um contrato local, certifique-se de que essa etapa tenha sido completada antes de liberar a conta. A verificação impedirá omissões e reforçará a base legal de quaisquer sanções que você tenha de impor no futuro.

Certifique-se de lembrar os novos usuários para alterarem suas senhas imediatamente. (Se desejar, você pode obrigá-los a fazer isso configurando para que esta senha inicial se expire num curto período de tempo. Uma outra opção é a de verificar os novos usuários e garantir que suas senhas criptografadas no arquivo **shadow** foram modificadas.)

Registrando a situação do usuário e informações para contato

Num ambiente em que você conhece todos os usuários pessoalmente é de certo modo fácil controlar quem está usando o sistema e o motivo para tal. Mas se você gerencia uma grande e mutável base de usuários, precisará de uma maneira mais formal de controlar essas contas. Manter um banco de dados contendo informações para contato e situações das contas o ajudará a descobrir quem é quem e por que tem a conta, uma vez que o processo de adição não está mais na memória. É uma boa idéia manter informações de contato completas à mão para que possa localizar os usuários na eventualidade de problemas ou mau comportamento.

6.5 Eliminando Usuários

Quando um usuário deixa uma empresa, a conta deste usuário e seus respectivos arquivos devem ser eliminados do sistema. Esse procedimento envolve a eliminação de todas as referências ao nome de login que foram adicionadas por você ou por seu programa **useradd**. O equivalente automatizado de **useradd** é **userdel**. Se você eliminar um usuário manualmente, talvez queira usar a seguinte lista de verificação:

- Configurar a cota de disco deste usuário em zero, caso o sistema de cotas esteja sendo utilizado.
- Eliminar o usuário de todos os bancos de dados e listas telefônicas de usuários.
- Eliminar o usuário do arquivo **aliases** ou adicionar um endereço de encaminhamento.
- Eliminar o arquivo crontab do usuário e quaisquer tarefas **at** pendentes.
- Extinguir qualquer processo do usuário que ainda esteja em execução.
- Eliminar quaisquer arquivos temporários de propriedade deste usuário em **/var/tmp** ou **/tmp** (um tanto retentiva, porém útil em alguns casos).
- Eliminar o usuário dos arquivos **passwd**, **shadow** e **group**.
- Eliminar o diretório inicial do usuário.
- Eliminar o spool de e-mail do usuário.

Antes de eliminar o diretório inicial de um usuário, certifique-se de realocar quaisquer arquivos que sejam necessários a outros usuários. Já que normalmente não podemos ter certeza de quais arquivos possam ser estes, é sempre uma boa idéia criar uma fita de backup extra do diretório inicial do usuário e de spool de e-mail antes de eliminá-los.

Assim que tiver eliminado um usuário, talvez seja interessante verificar se o UID antigo do usuário não tem propriedade sobre outros arquivos no sistema. Para localizar os caminhos de arquivos órfãos, pode-se usar o comando **find** com o argumento **-nouser**. Pelo fato de **find** ter uma maneira de "escapar" em servidores de redes caso você não seja cauteloso, é melhor verificar os sistemas de arquivos individualmente com **-xdev**:

```
# find filesystem -xdev -nouser
```

Se sua empresa atribui estações de trabalho individuais aos usuários, geralmente é mais fácil e mais eficiente reinstalar todo o sistema por meio de um gabarito mestre antes de passar o sistema a um novo usuário. Porém, antes de reinstalar é uma boa idéia fazer backup de arquivos locais do disco rígido do sistema no caso de eles virem a ser necessários no futuro.

6.6 Desabilitando Logins

De vez em quando, o login de um usuário precisa ser temporariamente desabilitado. Antes de as redes invadirem o mundo da computação, simplesmente colocávamos um asterisco em frente da senha criptografada, tornando impos-

www.snow666.blogspot.com
Snow666

sível ao usuário efetuar o login. Porém, os usuários ainda poderiam entrar no sistema através da rede sem ter de introduzir uma senha; ou seja, essa técnica não funciona mais de maneira apropriada.

Nos dias de hoje, substituímos o shell do usuário por um programa que imprime uma mensagem explicando o porquê do login ter sido desabilitado, fornecendo instruções para retificar essa situação. Esse pseudo-shell não deve ser listado em **/etc/shells**; muitos daemons que fornecem acesso não login ao sistema (por exemplo, **ftpd**) fazem uma verificação para ver se o shell de login de um usuário se encontra em **/etc/shells** e negará acesso caso ele não esteja presente (que é o comportamento que desejamos). Infelizmente, essa mensagem provavelmente não será vista caso os logins em sua instalação ocorram tipicamente em um sistema com janelas.

Entretanto, há um problema com este método de desabilitação de logins. Por padrão, **sendmail** não encaminhará mensagens a um usuário cujo shell não apareça em **/etc/shells**. Geralmente é uma má idéia interferir no fluxo de e-mail, mesmo se o receptor não estiver habilitado a lê-lo imediatamente. Podemos anular o comportamento-padrão de **sendmail** adicionando um shell falso chamado **/SENDMAIL/ANY/SHELL/** ao arquivo **/etc/shells** (embora possam ocorrer efeitos colaterais indesejados ao fazermos isso).

6.7 Utilitários para Gerenciamento de Contas

O comando **useradd** adiciona usuários ao arquivo **passwd** (e ao arquivo **shadow** se aplicável). Ele fornece uma interface dirigida por linhas de comando que é fácil de ser executada manualmente ou de chamar de um script **adduser** caseiro. O comando **usermod** altera as entradas **passwd** de usuários existentes. O comando **userdel** elimina um usuário do sistema, deletando opcionalmente o diretório inicial do usuário. Os comandos **groupadd**, **groupmod** e **groupdel** também operam sobre o arquivo **/etc/group**.

Por exemplo, para criarmos um usuário chamado "hilbert" com o comando **useradd** (usando os padrões do sistema), poderíamos simplesmente fazer o seguinte:

```
# useradd hilbert
```

Esse comando criaria a seguinte entrada em **/etc/passwd**. Note que **useradd** coloca um asterisco no campo de senha, desabilitando efetivamente a conta até que atribuamos uma senha real.

```
hilbert:*:105:20::/home/hilbert:/bin/sh
```

Esteja atento: se você não especificar explicitamente os grupos para o novo usuário, esse usuário será colocado num grupo-padrão (normalmente "users"), porém o arquivo **/etc/group** não será atualizado para incluir o nome deste novo usuário.

useradd geralmente é mais útil quando são usados argumentos adicionais. No próximo exemplo, especificamos o grupo primário de hilbert como "faculty" e que ele também deve ser adicionado ao grupo "famous". Também sobrepomos a localização de diretório inicial padrão e solicitamos a **useradd** que crie o diretório inicial caso ele ainda não exista:

```
# useradd -c "David Hilbert" -d /home/math/hilbert -g faculty -G famous
      -m -s /bin/tcsh hilbert
```

Esse comando cria a seguinte entrada **passwd**:

```
hilbert:x:1005:30:David Hilbert:/home/math/hilbert:bin/tcsh
```

(o UID atribuído é uma unidade maior do que o UID mais elevado do sistema) e a entrada **shadow** correspondente:

```
hilbert:!:11508:0:99999:7:0::
```

Ele também acrescenta hilbert aos grupos "faculty" e "famous" em **/etc/groups**, cria o diretório **/home/math/hilbert** e o preenche de acordo com o conteúdo do diretório **/etc/skel**.

Podemos determinar os ajustes de configuração-padrão de **useradd** através de **useradd -D**; também podemos usar o flag **-D** em combinação com outros argumentos para configurar estes padrões. Os padrões são armazenados em **/etc/default/useradd** e podem ser editados diretamente caso prefira.

usermod modifica uma conta que já existe e usa vários dos mesmos flags de **useradd**. Poderíamos, por exemplo, usar o comando a seguir para configurar uma data de expiração igual a 4 de julho de 2003 na conta de hilbert:

```
# usermod -e "July 4, 2003" hilbert
```

Snow666

O comando **userdel** elimina contas de usuário, desfazendo efetivamente todas as alterações feitas por **useradd**. Para eliminar hilbert, usaríamos o seguinte comando:

```
# userdel hilbert
```

Esse comando remove referências a hilbert nos arquivos **passwd**, **shadow** e **group**. Por padrão, ele não eliminaria o diretório inicial de hilbert.[7] A opção **-r** faz com que **userdel** também elimine o diretório inicial do usuário, porém mesmo em seu modo mais agressivo, **userdel** ainda executa somente as três últimas tarefas da lista de "obrigações de eliminação de usuário".

Embora os comandos **useradd** e **userdel** sejam convenientes, eles normalmente não são suficientes para implementar as políticas locais de uma instalação. Não hesite em criar seus próprios scripts **adduser** e **rmuser**; a maioria das grandes instalações faz isso. (Perl geralmente é a ferramenta indicada para essa tarefa.) Seus scripts caseiros podem chamar os utilitários-padrão para realizarem parte de suas tarefas.

6.8 EXERCÍCIOS

E6.1 Como é determinado o grupo-padrão de um usuário? Como poderíamos modificá-lo?

E6.2 Explique as diferenças entre os seguintes valores **umask**: 077, 027, 022 e 755. Como você tentaria implementar um desses valores como um padrão válido para todo o sistema no caso de novos usuários? É possível impor um padrão **umask** a seus usuários?

E6.3 Qual é o objetivo do arquivo de senhas-sombra?

★ **E6.4** Liste as etapas necessárias para adicionar um usuário a um sistema sem usar o comando **useradd**. Que etapas adicionais seriam necessárias para o seu ambiente local?

★ **E6.5** Determine a convenção de atribuição de nomes para novos usuários existente em sua instalação. Quais são as regras? Como a exclusividade é preservada? Você seria capaz de imaginar algum inconveniente? Como os usuários são eliminados?

★★ **E6.6** Localize uma lista de nomes de estudantes (de uma lista telefônica on-line, quem sabe) e use-a como entrada para um script que forma nomes de login de acordo com a convenção de atribuição de nomes de sua instalação. Quantos usuários podem ser acomodados antes de ser provocado um conflito? Quantos conflitos existem no todo? Use esses dados para avaliar a convenção de atribuição de nomes de sua instalação e sugira melhorias.

★★ **E6.7** Crie um script para ajudar a monitorar a saúde de seu arquivo **/etc/passwd**. (Os itens b e e requerem acesso root a menos que você seja esperto.)

a) Localize quaisquer entradas que possuam UID 0.
b) Localize quaisquer entradas que não possuam nenhuma senha (necessário **/etc/shadow**).
c) Localize quaisquer entradas que possuam UIDs duplicados.
d) Localize quaisquer entradas que possuam nomes de login duplicados.
e) Localize quaisquer entradas que não possuam nenhuma data de expiração (necessário **/etc/shadow**).

★★★★★ **E6.8** Projete um esquema para um banco de dados LDAP que armazene informações de usuário como login, senha, shell, máquinas autorizadas etc. Construa uma ferramenta que introduza interativamente novos usuários no banco de dados ou os obtenha de uma lista de usuários a serem criados. Crie uma ferramenta para as máquinas de seu laboratório que gere os arquivos **passwd**, **group** e **shadow** do banco de dados LDAP. Permita aos estudantes terem senhas diferentes em cada máquina que quiserem. (Nem todos os estudantes possuem necessariamente autorização para usar todos os computadores.) Seu sistema **adduser** poderá imprimir listas de nomes de login para instrutores e imprimir pares login/senha para usuários novos.

7. Em nossa instalação geralmente preservamos os diretórios iniciais de usuários eliminados por algumas semanas. Essa política minimiza a necessidade de restauração de dados por meio de fitas de backup caso um usuário eliminado retorne ou se outros usuários precisarem ter acesso aos arquivos de trabalho do usuário eliminado.

Snow666

7 Dispositivos Seriais

Portas seriais é um dos dispositivos de E/S mais utilizáveis disponíveis nos sistemas Linux. Apesar de não serem especialmente rápidas, são flexíveis e baratas.

As portas seriais podem ser usadas com uma série de dispositivos, inclusive impressoras, terminais e outros computadores. Elas também são encontradas numa ampla gama de equipamentos personalizados, de hobistas e de baixa voltagem (trocadores automáticos de mídia, sensores de temperatura e até mesmo máquinas de costura). Um dispositivo pode ser conectado ao sistema diretamente (por meio de um cabo) ou via linha telefônica com modems nas duas pontas.

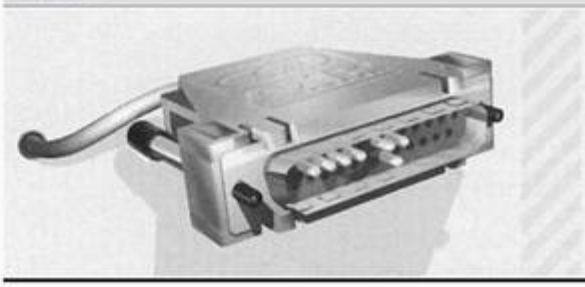
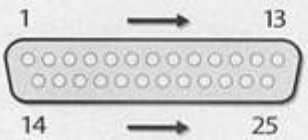
Este capítulo descreve como conectar dispositivos a seu sistema e explica como configurar seu software para tirar proveito deles. Normalmente usaremos modems e impressoras como exemplos específicos, porém outros dispositivos são basicamente similares.

As primeiras seções tratam de considerações de cabeamento e de hardware serial. A seguir, falamos sobre a infra-estrutura de software que tem sido historicamente usada para dar suporte a terminais de hardware. Os terminais raramente são usados hoje em dia, porém seus fantasmas sobrevivem na manipulação do Linux de pseudoterminais e sistemas com janelas. O restante do capítulo fornece algumas informações gerais sobre modems, depuração de dispositivos seriais, portas paralelas e USB (Universal Serial Bus, barramento serial universal).

7.1 Padrões de Dispositivos Seriais

A maioria das portas seriais obedece a alguma variante do padrão conhecido como RS-232. Esse padrão especifica as características elétricas e o significado de cada pino de sinal, bem como atribuições de pinagem no conector serial (DB-25) de 25 pinos tradicional conforme mostrado no Diagrama A.

Diagrama A Um conector DB-25 macho

Conector	Numeração dos pinos
	

Snow666

O padrão RS-232[1] completo é excessivamente grande na maioria das situações práticas, já que ele define numerosos sinais que não são necessários para comunicação básica. Os conectores DB-25 também são inconvenientemente grandes. Como resultado, uma série de conectores alternativos acabou se tornando largamente usada. Estes são descritos na seção *Conectores Alternativos* começando na página 79.

O RS-232 tradicional usa cabo blindado de par trançado (STP), normalmente fios trançados 22[2]. As voltagens de sinais do RS-232 original eram de ±12 Volts DC, porém ±5 Volts é mais comum hoje em dia. Algumas vezes é usado ± 3 Volts. Voltagens maiores são menos suscetíveis à interferência. Todas essas voltagens atendem à especificação RS-232, de forma que é perfeitamente correto conectar dispositivos que usem padrões de voltagem diferentes.

O RS-232 não é um sistema eletricamente "balanceado"; ele usa um único condutor para os dados que trafegam em cada direção. Portanto, as propriedades elétricas especiais de cabos de par trançado podem ser menos significativas para comunicação serial do que são para, digamos, redes Ethernet. De fato, os cabos de par trançado podem realmente reduzir a confiabilidade e a amplitude de uma conexão serial caso as duas linhas de dados (TD e RD) forem colocadas juntas num único par. Portanto, não faça isso.

Não há nenhum padrão sobre quais sinais RS-232 deveriam percorrer juntos num cabo de par trançado. Algumas fontes recomendam emparelhar sinais de aterramento com TD e RD, porém este emparelhamento significa um condutor adicional e fornece vários caminhos para o sinal de terra. Pelo que nos consta, não há nenhuma razão que force o uso dessa convenção.

Os conectores DB-25 podem ser ou machos (com os pinos indo para fora, chamados de DB25P) ou fêmeas (com orifícios concordantes com os pinos, chamados de DB25S). Há minúsculos números quase invisíveis próximos dos pinos ou orifícios numerando-os de 1 a 25. Podemos ver melhor os números colocando o conector contra a luz e observando-o de maneira enviesada. Algumas vezes somente os pinos 1, 13, 14 e 25 são numerados.

O Diagrama A mostra um conector DB-25 macho. Como acontece com todos os conectores seriais, os números de pinos num conector fêmea representam uma imagem espelhada daqueles contidos num conector macho para que pinos de mesma numeração se encaixem. O diagrama é desenhado com base na orientação mostrada (como se estivéssemos olhando para o final do cabo, prestes a plugar o conector em sua parte frontal).

Note que, no Diagrama A, somente sete pinos são realmente instalados. Isso é típico para o mundo real. Os sinais RS-232 e suas atribuições de pinos num conector DB-25 são mostrados na Tabela 7.1. Somente os sinais sombreados são usados na prática (pelo menos em sistemas computacionais genéricos); todos os demais podem ser ignorados.

Tabela 7.1 Sinais RS-232 e atribuições de pinos num conector DB-25[3].

Pino	Nome	Função	Pino	Nome	Função
1	FG	Terra de frame	14	STD	TD secundário
2	TD	Dados transmitidos	15	TC	Relógio para transmissão
3	RD	Dados recebidos	16	SRD	RD secundário
4	RTS	Sinal de solicitação de envio	17	RC	Relógio para recebimento
5	CTS	Transmissão pode prosseguir	18	–	Não empregado
6	DSR	Pronto para dados	19	SRTS	RTS secundário
7	SG	Terra de sinal	20	DTR	Pronto para receber uma transmissão
8	DCD	Detecção de portadora de dados	21	SQ	Detector de qualidade de sinal
9	–	Tensão positiva	22	RI	Indicador de anel
10	–	Tensão negativa	23	DRS	Seletor de velocidade de dados
11	–	Não empregado	24	SCTE	Relógio de transmissão externo
12	SDCD	DCD secundário	25	BUSY	Ocupado
13	SCTS	CTS secundário			

1. Para ser tecnicamente correto, este padrão deveria ser chamado de EIA-232-E. Entretanto, ninguém terá a mínima idéia sobre o que você está falando.
2. Esse número se refere à métrica AWG (American Wire Gauge); no Brasil, o equivalente seria um fio com seção nominal 0.3 mm (N.R.T.).
3. A revisão D do padrão RS-232 alterou a nomenclatura e a utilização do Pino 1, passando para PG (Protective Ground, terra de proteção), de modo que forneça uma blindagem eficiente. Esse pino normalmente deve ser conectado internamente à carcaça do dispositivo (N.R.T.).

www.snow666.blogspot.com
Snow666

Há duas configurações de interfaces para equipamentos seriais: DTE (Data Terminal Equipment) e DCE (Data Communications Equipment). DTE e DCE compartilham as mesmas pinagens, porém eles especificam interpretações diversas dos sinais RS-232.

Cada dispositivo é configurado como DTE ou como DCE, mas não os dois simultaneamente. Computadores, terminais e impressoras geralmente são DTE, e a maioria dos modems é DCE. As portas seriais DTE e DCE podem se comunicar entre si em qualquer combinação, mas combinações diversas exigem cabeamentos diferentes.

Não há nenhuma razão lógica para existirem tanto o DTE como o DCE; todos os equipamentos poderiam usar o mesmo esquema de fiação. A existência das duas convenções é meramente um dos diversos legados históricos sem sentido do RS-232.

DTE e DCE podem se tornar relativamente confusos caso você se permita ficar pensando muito no assunto. Quando isso acontece, simplesmente respire fundo e releia os seguintes pontos:

- A pinagem do RS-232 para um determinado tipo de conector é sempre a mesma, independentemente de o conector ser macho ou fêmea (números de pinos concordantes sempre coincidem) e independentemente de o conector estar num cabo, num dispositivo DTE ou num dispositivo DCE.
- Toda a terminologia RS-232 se baseia no modelo de uma conexão direta de um dispositivo DTE para um dispositivo DCE. (Por "direta" queremos dizer que TD no DTE é conectado a TD no DCE e assim por diante. Cada pino se conecta com o pino de mesma numeração do outro lado.)
- Os sinais recebem nomes sob a perspectiva do dispositivo DTE. Por exemplo, o nome TD (Transmitted Data, dados transmitidos) realmente significa "dados transmitidos do DTE para o DCE". Independentemente do nome, o pino TD é uma entrada num dispositivo DCE. De forma similar, RD é uma entrada para o DTE e uma saída para o DCE.
- Ao conectarmos equipamento DTE a equipamento DTE (computador-terminal ou computador-computador), temos de iludir cada dispositivo fazendo com que eles acreditem que o outro é o DCE. Por exemplo, ambos os dispositivos DTE têm a expectativa de transmitir em TD e receber em RD; temos de conectar os fios de maneira invertida para que o pino de transmissão de um dispositivo vá para o pino de recepção do outro dispositivo e vice-versa.
- Três conjuntos de sinais devem ser cruzados desta maneira para comunicação DTE-DTE (se optarmos por conectá-los todos). TD e RD devem ser cruzados. RTS e CTS devem ser cruzados. E o pino DTR de cada lado tem de estar conectado tanto ao pino DCD como ao DSR do correspondente.
- Para confundir ainda mais, um cabo cruzado para comunicação DTE-DTE é normalmente chamado de cabo "modem nulo". Você pode ser tentado a usar um cabo modem nulo para conectar um modem, porém, como os modems são DCE, isso não funcionará! Um cabo para modem é chamado de "cabo modem" ou "cabo direto".

Devido à questão DTE *versus* DCE ser tão confusa, pode ser que ocasionalmente você veja tentativas bem-intencionadas, porém incorretas, de dar alguma luz à nomenclatura definindo DTE e DCE como se tivessem pinagens distintas (por exemplo, renomeando o pino TD de um DCE para RD e vice-versa). Nesse universo alternado, as pinagens variam, mas as conexões de cabos (por nome de sinal) não variam. Sugerimos que você ignore qualquer material que fale sobre "pinagem DTE" ou "pinagem DCE"; é pouco provável ser uma fonte de informação confiável.

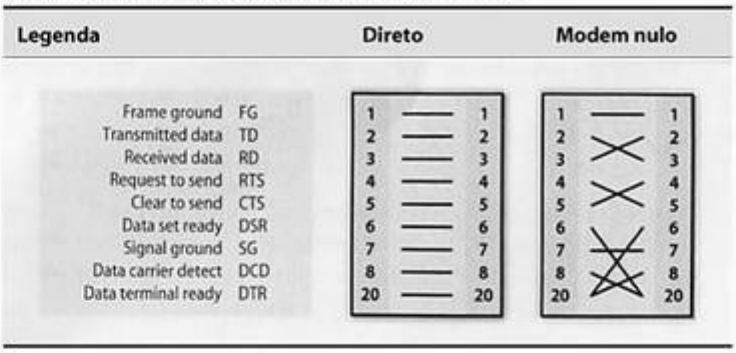
Originalmente, os dispositivos DTE deveriam supostamente ter conectores machos e os dispositivos DCE serem conectores fêmea. Por acaso, os projetistas de hardware descobriram que os conectores machos eram mais frágeis. Hardware computacional caro normalmente possui agora conectores fêmea e a maioria dos cabos é machos em ambas as extremidades. [4]

O Diagrama B da próxima página mostra as atribuições de pinos e conexões tanto para cabos diretos quanto para cabos modem nulo. São apresentados somente os sinais usados na prática.

4. Na Qwest, os termos "macho" e "fêmea" são considerados inadequados. Os funcionários são incentivados a usar os termos "plugue" e "receptáculo". Os nomes de conector padrão DB25P e DB25S na verdade derivam de uma terceira convenção: "pino" e "soquete".

Snow666

Diagrama B Atribuições de pinos e conexões para cabos DB-25



7.2 Conectores Alternativos

As seções a seguir descrevem a maioria dos sistemas alternativos de conectores: mini DIN-8, DB-9 e RJ-45. Apesar de suas diferenças físicas, todos estes conectores fornecem acesso aos mesmos sinais elétricos do que um conector DB-25. Dispositivos que usam conectores diferentes sempre são compatíveis quando o tipo correto de cabo conversor é utilizado.

A variante mini DIN-8

Os conectores mini DIN-8 são encontrados em Macs e em alguns laptops e estações de trabalho. Este conector quase que circular e extremamente compacto fornece conexões para sete sinais. Ele é ilustrado no Diagrama C.

Diagrama C Um conector mini DIN-8 macho

Lojas de informática normalmente vendem cabos conversores DB-25 injetados por molde para mini DIN 8. Não tente você mesmo confeccioná-los, pois um mini DIN-8 é tão pequeno que é um desafio estabelecer-se conexões seguras usando as mãos. As atribuições de pinos são mostradas na Tabela 7.2

Tabela 7.2 Pinos para um cabo direto mini DIN-8 para DB-25.

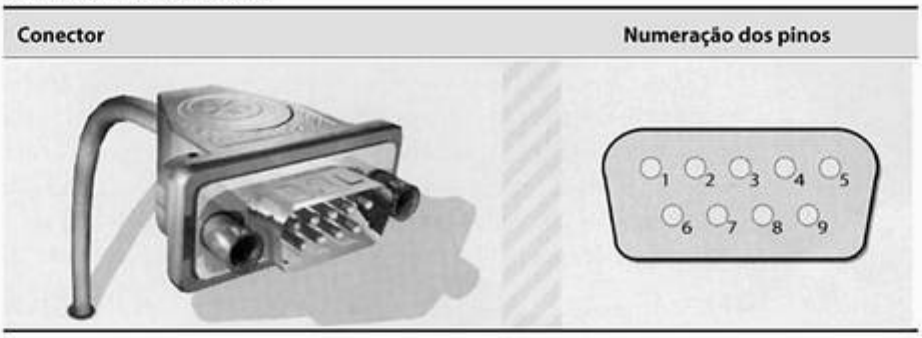
DIN-8	DB-25	Sinal	Função
3	2	TD	Dados transmitidos
5	3	RD	Dados recebidos
6	4	RTS	Sinal de solicitação de envio
2	5	CTS	Transmissão pode prosseguir
4,8	7	SG	Terra de sinal
7	8	DCD	Detecção de portadora de dados
1	20	DTR	Pronto para receber uma transmissão

A variante DB-9

Encontrado comumente em PCs, este conector de nove pinos (que se parece com um conector DB-25 "júnior") fornece os oitos sinais mais comumente usados.

Snow666

Diagrama D Um conector DB-9 macho



Lojas de informática em sua região devem ter disponível cabos conversores DB-9 para DB-25 pré-fabricados. A Tabela 7.3 mostra as atribuições de pinos.

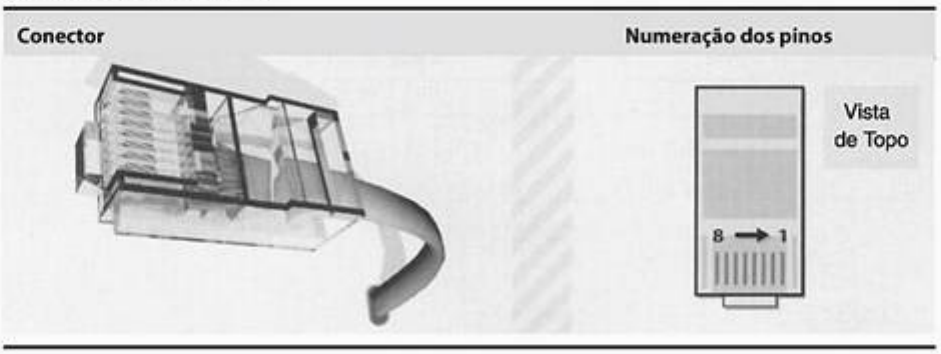
Tabela 7.3 Pinos para um cabo direto DB-9 para DB-25.

DB-9	DB-25	Sinal	Função
3	2	TD	Dados transmitidos
2	3	RD	Dados recebidos
7	4	RTS	Sinal de solicitação de envio
8	5	CTS	Transmissão pode prosseguir
6	6	DSR	Pronto para dados
5	7	SG	Terra de sinal
1	8	DCD	Detecção de portadora de dados
4	20	DTR	Pronto para receber uma transmissão

A variante RJ-45

RJ-45 é um conector telefônico modular de oito fios. Ele é similar ao conector RJ-11 padrão usado para fiação telefônica nos Estados Unidos, porém um conector RJ-45 possui oito fios (um conector RJ-11 possui apenas quatro[5]). Os conectores RJ-45 são mais comumente usados para fiação Ethernet, mas eles também funcionam bem em comunicação serial.

Diagrama E Um conector RJ-45 macho



Os plugues RJ-45 normalmente não são encontrados em computadores ou equipamentos seriais diversos, porém eles são freqüentemente usados como conectores intermediários para rotear linhas seriais através de painéis de conexão. Eles podem algumas vezes ser encontrados em dispositivos que têm muitas portas próximas como, por exemplo, em servidores de terminais. Os RJ-45s são algumas vezes usados com cabo telefônico chato em vez de pa-

5. Uma análise cuidadosa revelará que um plugue RJ-11 na verdade possui seis aberturas onde poderíamos ter pinos, mas que somente quatro condutores estão instalados. Um verdadeiro conector telefônico de quatro condutores é um RJ-10. Um RJ-11 com todos os seis pinos instalados é conhecido como RJ-12.

Snow666

res trançados em feixe. Ambas as formas de cabos são aceitáveis para conexões seriais embora o par trançado provavelmente ofereça melhor qualidade de sinal em distâncias mais longas.

Os RJ-45s são compactos, auto-seguros e baratos. Eles são grampeados no cabo com uma ferramenta especial. Leva menos do que um minuto para anexar um. Caso esteja projetando um sistema de cabeamento grande da estaca zero, os RJ-45s são uma boa opção para conectores intermediários.

Há vários sistemas para associar os pinos num conector RJ-45 àqueles de um DB-25. A Tabela 7.4 mostra o padrão RS-232D oficial, que é usado somente casualmente.

Uma maneira alternativa de empregar RJ-45s é por meio do sistema de Dave Yost, que adiciona um soquete RJ-45 para cada dispositivo e usa somente um único tipo de cabo de conector RJ-45 independentemente de os dispositivos serem DTEs ou DCEs. O sistema de Dave é mais do que simplesmente um plugue com pinos; ele é um sistema de cabeamento completo que atende a vários problemas diferentes. O padrão de Dave é apresentado na próxima seção.

Esta especificação foi escrita por Dave Yost (Dave@Yost.com).

O padrão Yost para cabeamento RJ-45

Eis um esquema que oferece soluções para vários problemas envolvendo RS-232:

- Todos os conectores de cabos devem ser do mesmo tipo e sexo (macho RJ-45).
- Não há nenhuma distinção entre DTE e DCE.

Tabela 7.4 Pinos para um cabo direto RJ-45 para DB-25.

RJ-45	DB-25	Sinal	Função
1	6	DSR	Pronto para dados
2	8	DCD	Detecção de portadora de dados
3	20	DTR	Pronto para receber uma transmissão
4	7	SG	Terra de sinal
5	3	RD	Dados recebidos
6	2	TD	Dados transmitidos
7	5	CTS	Transmissão pode prosseguir
8	4	RTS	Sinal de solicitação de envio

- Precisamos apenas de um tipo de cabo conector.
- Podemos produzir cabos em massa rapidamente, usando somente uma ferramenta para grampeamento.

Cada porta serial em cada equipamento obtém seu adaptador DB-25 ou DB-9 para RJ-45 com a fiação apropriada. Esse adaptador é permanentemente atarraxado à porta. A porta agora apresenta a mesma interface de conector, fêmea RJ-45, independentemente de seu conector implícito ser DB-25 ou DB-9, DTE ou DCE, macho ou fêmea. Além disso, toda porta serial agora transmite e recebe dados nos mesmos pinos.

Assim que tiver colocado esses adaptadores em suas portas RS-232, poderá conectar qualquer coisa com qualquer coisa sem usar modems nulos ou terminais nulos, mudar pinos em conectores de cabos ou construir cabos especiais. Podemos conectar um modem a um computador, modem a terminais, terminais a computadores, terminais a terminais, computador a computador etc., tudo isso com apenas um tipo de cabo.

Os cabos são cabos de fita de oito fios com jackets. Os conectores em cada extremidade são espremidos no cabo com uma ferramenta de grampeamento, de forma que não há nenhum tipo de soldagem ou de confusão com os pinos.

Há três fios de sinais (um de dados e dois de controle) indo em cada direção, mais um par de terras de sinal. Os cabos não são conectados normalmente (isto é, com cada pino do conector ligado ao pino correspondente no outro lado do cabo). Eles são ligados "com um entrançamento" ou "imagem espelhada" "ou invertidos lado a lado", ou seja lá o que você denomine. Isto é, o pino 1 no final do cabo se liga ao pino 8 na outra extremidade etc. (Esse significado de "entrançamento" é distinto daquele usado no termo "par trançado".) Esse esquema funciona porque o layout dos sinais no cabo de fita é simétrico. Isto é, cada pino de transmissão possui o seu pino de recepção correspondente na posição de fio de imagem espelhada através do cabo plano. [6]

6. Dave não diz explicitamente, porém temos de efetivamente ligar o cabo sem um entrançamento físico para obtermos o efeito "com entrançamento". Pelo fato de os conectores nas extremidades de um cabo apontarem para direções opostas entre eles, a numeração de pinagem é invertida automaticamente.

Snow666

Cabos RJ-45 já prontos são normalmente ligados diretamente. Para usá-los com esse sistema você terá de remover o conector de uma extremidade e grampear numa nova extremidade com os fios invertidos. Conectores RJ-45 fêmea-fêmea são disponíveis para estender cabeamento, porém lembre-se do seguinte: dois cabos trançados unidos com tal acoplador criam um cabo direto.

Muitos fornecedores fabricam adaptadores DB-25 para RJ-45. A codificação de cores interna não coincide com as cores do cabo. Os adaptadores, conectores de fios e fiação se tornaram disponíveis nas lojas de eletrônica, infelizmente sem qualquer ajuda sobre como usá-los para RS-232.

Veja a seção 15.2, Capítulo 15, para obter maiores informações sobre o cabo Categoria 5.

Esse esquema destinava-se para uso com cabo de fita com jacket, no qual todos os fios estão lado a lado. Os cabos de par trançado, por contraste, possuem quatro pares de fios, cada par trançado em torno de si próprio ao longo do comprimento do cabo. Se você usa cabos de par trançado (como o cabo Categoria 5), não deve ligar seus cabos como normalmente faria para o RJ-45 (por exemplo, para 10BaseT, telefônico etc.). Em vez disso, você deve ligá-los de modo que os fios 3:4 e fios 5:6 formem pares. Outros emparelhamentos serão suscetíveis à interferência de sinal de dados. O emparelhamento dos fios restantes não é importante, porém 1:2 e 7:8 serão tão bons quanto qualquer outro que você queira.

Dentro de um adaptador temos um soquete RJ-45 com oito fios provenientes dele. Esses fios têm pinos RS-232 (ou soquetes de pinos, conforme apropriado) grampeados neles. Simplesmente empurramos esses pinos para os orifícios apropriados no conector RS-232 e depois fechamos o alojamento do adaptador.

Tabela 7.5 Fiação para um adaptador RJ-45 para DB-25 ou DB-9 Yost.

			Conectar a pinos DCE			Conectar a pinos DTE		
Cabo RJ-45		**Adaptador**	**DB-25**	**DB-9**	**Sinal**	**DB-25**	**DB-9**	**Sinal**
1	Marrom (com Cinza)	Azul	4	7	RTS	5	8	CTS
2	Azul (com Laranja)	Laranja	20	4	DTR	8	1	DCD
3	Amarelo (com Preto)	Preto	2	3	TD	3	2	RD
4	Verde (com Vermelho)	Vermelho	7	5	GND	7	5	GND
5	Vermelho (com Verde)	Verde	7	5	GND	7	5	GND
6	Preto (com Amarelo)	Amarelo	3	2	RD	2	3	TD
7	Laranja (com Azul)	Marrom	8	1	DCD	20	4	DTR
8	Cinza (com Marrom)	Branco	5	8	CTS	4	7	RTS

Contudo, há um problema: ambos os pinos-terra têm de se encaixar no mesmo orifício DB-25 ou DB-9 (pino 7 ou 5, respectivamente). Você pode grampear esses fios juntos para que eles venham a formar um único pino com um pequeno treco plástico fabricado pela AMP chamado de "Tel-splice connector ½ tap dry", número da peça 553017-4. Até então, essa peça parece poder ser encontrada somente em grande quantidade (1.000 peças) a US$ 80 mais ou menos. Acredite-me, você vai querer usá-las caso vá ligar um número razoável de adaptadores.

Alguns dispositivos DTE requerem que o sinal DSR esteja ativo antes de eles enviarem dados. Esse sinal é usualmente fornecido pelo dispositivo DCE, porém você pode simulá-lo ligando os pinos 20 e 6 (4 e 6 num conector DB-9). Dessa maneira, o dispositivo DTE receberá o sinal DSR dele próprio sempre que ele clamar DTR.

Em algumas impressoras DCE, o pino 7 do adaptador RJ-45 (o fio marrom) deve ser conectado à linha DSR (pino 6 tanto no DB-25 quanto no DB-9). Leia a documentação de sua impressora para descobrir se ela fornece sinais de handshaking úteis no DSR em vez de DCD.

Dave Yost Los Altos, CA julho de 1999.

Não use nenhuma dessas conexões DSR como uma seqüência lógica e natural; adicione-as somente a dispositivos específicos que pareçam precisar delas (ou que não pareçam operar com a configuração-padrão). Alguns dispositivos "comuns" toleram as conexões extras, porém outros ficam confusos.

7.3 Portadora via Hardware e Software

O Linux espera que o sinal DCD, detecção de portadora, fique alto (tensão positiva) quando um dispositivo é conectado e ligado. Esse sinal é conduzido no pino 8 do conector DB-25 padrão. Se seu cabo serial tiver uma linha DCD e seu computador realmente prestar atenção a isto, você estará usando o que é conhecido como uma portadora via hardware. A maioria dos sistemas também permite portadora via software, em que o computador simula que o DCD é sempre afirmado.

Snow666

Para determinados dispositivos (particularmente terminais), a portadora via software é uma grande dádiva. Ela permite que você se safe usando somente três linhas para cada conexão serial: transmitir, receber e terra de sinal. Entretanto, as conexões de modem realmente precisam do sinal DCD. Se um terminal está conectado através de um modem e o sinal de portadora for perdido, o modem deve suspender (especialmente numa chamada interurbana!).

Normalmente especificamos portadora via software para uma porta serial no arquivo de configuração para qualquer software-cliente que estivermos usando em conjunto com a porta (por exemplo, **/etc/gettydefs** ou **/etc/inittab** para um terminal de login ou **/etc/printcap** para uma impressora). Também podemos usar **stty -clocal** para habilitar portadora via software em tempo de execução. Por exemplo,

```
# stty -clocal < /dev/ttyS1
```

habilitaria a portadora via software para a porta **ttyS1**.

7.4 Controle de Fluxo via Hardware

Os sinais CTS e RTS são usados para garantir que um dispositivo não envie dados mais rápido do que o receptor é capaz de processar. Por exemplo, se um modem está prestes a estourar o seu espaço de buffer (quem sabe porque a conexão com a instalação remota é mais lenta do que o link serial entre a máquina local e o modem), ele poderá informar o computador para não dialogar até que haja mais espaço disponível no buffer.

O controle de fluxo é essencial para modems de alta velocidade e também é muito útil para impressoras. Em sistemas que não suportam controle de fluxo via hardware (seja porque as portas seriais não o compreendam ou porque o cabo serial deixa os sinais CTS e RTS desconectados), o controle de fluxo pode algumas vezes ser simulado em software com os caracteres ASCII XON e XOFF. Entretanto, o controle de fluxo via software tem de ser explicitamente suportado por software de alto nível e mesmo assim ele não funciona bem.

XON e XOFF são, respectivamente, <Control-Q> e <Control-S>. Isto é um problema para usuários **emacs**, pois <Control-S> é a associação de teclas-padrão para o comando de pesquisa **emacs**.

Para solucionar tal problema, associe o comando de pesquisa a alguma outra tecla ou use **stty start** e **stty stop** para modificar o significado de XON e XOFF para o driver de terminal.

A maioria dos terminais ignora os sinais CTS e RTS. Fazendo um jumper entre os pinos 4 e 5 no final do cabo do terminal, podemos iludir os poucos terminais que exigem um handshake através destes pinos antes de eles se comunicarem. Quando o terminal envia um sinal no pino 4 dizendo "Estou pronto", ele obtém o mesmo sinal de volta no pino 5 dizendo "Vá em frente". Também podemos fazer um jumper do handshake DTR/DSR/DCD dessa maneira.

Como acontece com a portadora via software, o controle de fluxo via hardware pode ser manipulado por arquivos de configuração ou configurados via **stty**.

7.5 Comprimento do Cabo

O padrão RS-232 especifica um comprimento máximo de cabo de 22,5 m numa velocidade de 9.600 bps. Os padrões são normalmente conservadores e o RS-232 não é nenhuma exceção. Já usamos rotineiramente cabos RS-232 em distâncias muito maiores, até cerca de 300 m. Chegamos a um limite algo em torno de 240 a 300 m, porém descobrimos que a marca em particular dos dispositivos em cada ponta faz uma bela diferença.

Os drivers de linha ou repetidores são usados algumas vezes com RS-232 para se atingir o comprimento máximo de uma conexão. Infelizmente, esses dispositivos normalmente só amplificam os pinos RD e TD, para que outros sinais possam precisar ser jumpeados.

7.6 Arquivos de Dispositivos Seriais

As portas seriais são representadas pelos arquivos de dispositivos em ou sob **/dev**. A maioria dos computadores possui duas portas seriais: **/dev/ttyS0** e **/dev/ttyS1**. As distribuições do Linux normalmente vêm com um conjunto completo de arquivos de dispositivos (64 ou mais) para portas seriais adicionais pré-instaladas, porém até que adicionemos mais hardware ao sistema, os arquivos extras são supérfluos e devem ser ignorados.

Por razões históricas, as portas seriais primárias do sistema (**/dev/ttyS0** e **/dev/ttyS1**) também são conhecidas, respectivamente, como **/dev/cua0** e **/dev/cua1**. Os arquivos **cua*** eram usados antigamente para facilitar o gerenciamento de modems que manipulavam tanto chamadas de entrada como de saída. Entretanto, toda essa funcionalidade fornecida pelo sistema de dois arquivos agora se encontra disponível através dos arquivos **ttyS*** sozinhos, desde que o software-cliente saiba como solicitá-la. O emprego de dispositivos **cua*** hoje em dia é enfaticamente desencorajado, e o Debian nem mesmo os cria na situação-padrão.

Caso se depare com algum pacote de software antigo que não siga as atuais convenções, dê uma olhada no arquivo **doc/ttyS-cua.txt** no pacote **mgetty** para obter uma descrição do relacionamento existente entre arquivos **ttyS*** e **cua***.

www.snow666.blogspot.com
Snow666

Como sempre, os nomes dos arquivos de dispositivos realmente não importam. O comportamento é determinado pelos números de dispositivos principais e secundários e os nomes de dispositivos são meramente uma conveniência para nós seres humanos.

7.7 SETSERIAL: INFORME AO DRIVER SOBRE PARÂMETROS DE PORTAS SERIAIS

As portas seriais num PC podem ser configuradas para usar vários endereços de portas de I/O e níveis de interrupção (IRQs) diferentes. Essas configurações são normalmente acessadas através da BIOS do sistema no momento em que a máquina é ligada. A razão mais comum para alterá-las é a de acomodar algum hardware excêntrico que é intolerante em relação às suas próprias configurações e somente funcionará de forma correta quando tiver usurpado as configurações normalmente usadas por uma porta serial. Infelizmente, talvez o driver serial não seja capaz de detectar tais mudanças de configuração sem a sua ajuda.

A resposta de sistemas UNIX tradicionais a tal diversidade seria a de permitir que fossem especificados parâmetros de porta serial quando o kernel é compilado. Felizmente, o Linux permite que pulemos essa entediante etapa e modifiquemos os parâmetros durante a execução por meio do comando **setserial**. **setserial -g** exibe as configurações atuais:

```
# setserial -g /dev/ttyS0
/dev/ttyS0, UART: 16550A, Port: 0x03f8, IRQ: 4
```

Para configurarmos os parâmetros, especificamos o arquivo de dispositivos e depois uma série de parâmetros e valores. Por exemplo, o comando

```
# setserial /dev/ttyS1 port 0x02f8 irq 3
```

configura o endereço de porta de I/O e o IRQ para **ttyS1**. É importante ter em mente que esse comando não altera de maneira alguma a configuração de hardware; ele simplesmente informa o driver serial Linux da configuração. Para alterar realmente as configurações de hardware, consulte a BIOS de seu sistema.

setserial altera somente a configuração atual, e as configurações não persistem após reinicializações. Infelizmente, não há uma maneira-padrão de tornarmos permanentes as modificações; cada uma de nossas distribuições de exemplo o faz de forma diferente.

O script **/etc/rc.d/rc.sysinit** do Red Hat verifica se existe **/etc/rc.serial** e o executa no momento da inicialização caso ele exista. Não é fornecido nenhum modelo de arquivo, de modo que você mesmo tem de criar o arquivo caso queira usar este recurso. Simplesmente liste comandos **setserial** que deseja executar, um a cada linha. Para ser completo, provavelmente é uma boa idéia tornar o arquivo executável e colocar **#!/bin/sh** na primeira linha; entretanto, estes *toques de elegância* não são estritamente exigidos.

O SuSE fornece um script **/etc/init.d/serial** que trata da inicialização de portas seriais. Infelizmente, esse script não possui nenhum arquivo de configuração; você tem de editá-lo diretamente para refletir os comandos que deseja executar. SuSE malvado! O script usa sua própria pequena metalinguagem para construir as linhas de comando **setserial**, porém felizmente há uma série de linhas de exemplos comentadas para podermos escolher.

O Debian fornece um arquivo de configuração gentilmente comentado, **/etc/serial.conf**, que é lido por **/etc/init.d/setserial**. Ele fornece uma variedade de recursos avançados (como autoconfiguração persistente) que provavelmente é útil somente para os projetistas da distribuição. Para casos simples, basta retirar os comentários (#) das linhas que deseja ou então adicionar linhas próprias no formato **setserial** (omita o nome do comando).

7.8 CONFIGURAÇÃO DE SOFTWARE PARA DISPOSITIVOS SERIAIS

Assim que um dispositivo tiver sido conectado com o cabo apropriado, o software na máquina host deve ser configurado para tirar proveito dele. As tarefas de configuração para um dispositivo novo dependem do tipo de dispositivo e em que ele será empregado:

- Para um terminal hardwired ("terminal burro"), temos de informar o sistema para atender a logins na porta do terminal. Especificamos a velocidade e os parâmetros da conexão serial. A configuração para terminais é descrita na próxima seção.
- Modems de discagem são configurados de forma similar aos terminais hardwired. Entretanto, o procedimento exato pode ser ligeiramente diferente em alguns sistemas.

Snow666

- Para ver como usar um modem para se conectar a uma rede remota via PPP, consulte a página 218.
- Veja o Capítulo 24, *Impressão*, para obter maiores informações sobre como configurar uma impressora serial. Algumas impressoras somente recebem dados; outras são bidirecionais e podem retornar informações de estado para o computador host.
- Um dispositivo serial personalizado que usaremos somente por meio de software próprio não necessita de nenhuma configuração especial. Podemos simplesmente abrir o arquivo de dispositivos para acessar o dispositivo. Consulte a página de manual **termios** para aprender a configurar a velocidade, bits de flag e modo de buffering de uma porta serial.

7.9 Configuração de Terminais Hardwired

Ao longo da última década, computadores baratos substituíram praticamente todos os terminais ASCII. Entretanto, mesmo as janelas de "terminal" numa tela gráfica usam os mesmos drivers e arquivos de configuração do que os verdadeiros terminais, de forma que os administradores ainda precisam entender como eles funcionam.

A configuração de terminais envolve duas tarefas principais: certificar-se de que um processo está conectado a um terminal para aceitar logins e certificar-se que informações sobre o terminal estejam disponíveis assim que um usuário tiver logado. Entretanto, antes de mergulharmos de cabeça nos detalhes dessas tarefas, vamos dar uma olhada no processo de login como um todo.

*Veja o Capítulo 2 para obter maiores informações sobre o daemon **init**.*

O processo de login

O processo de login envolve vários programas diferentes, sendo o mais importante deles o daemon **init**. Uma das tarefas de **init** é gerar um processo, conhecido genericamente como **getty**, em cada porta de terminal que é ativada no arquivo **/etc/inittab**. **getty** configura as características iniciais da porta (como velocidade e paridade) e imprime um prompt de login.

O verdadeiro nome do programa **getty** varia entre as distribuições Linux, e algumas distribuições fornecem várias implementações. O Red Hat e o SuSE usam uma versão simplificada chamada **mingetty** para lidar com logins em consoles virtuais. Para o gerenciamento de terminais e modems de discagem, eles fornecem a implementação **mgetty** de Gert Doering. O Debian usa um getty único criado por Wietse Venema et al.; esta versão também é fornecida em sistemas SuSE sob o nome de **agetty**. Uma implementação mais antiga chamada **uugetty** foi suplantada de longe por **mgetty**.

Para fazer a distinção entre essa plenitude de **getty**s, imagine-os em ordem de complexidade. **mingetty** é o mais simples e é basicamente um substituto para **getty**. Ele é capaz de tratar logins somente em consoles Linux virtuais. **agetty** é um tanto mais encorpado e trata tanto de portas seriais como de modems. **mgetty** é o atual rei do pedaço. Ele manipula faxes que chegam, bem como logins, e executa o bloqueio e coordenação apropriados para que o mesmo modem possa ser usado tanto como linha discada de entrada como de saída.

A seqüência de eventos num login completo é a seguinte:

- **getty** imprime o conteúdo do arquivo **/etc/issue** juntamente com um prompt de login.
- Um usuário introduz um nome de login no prompt de **getty**.
- **getty** executa o programa **login** com o nome especificado como argumento.
- **login** solicita uma senha e a valida, comparando-a com a contida em **/etc/shadow**.[7]
- **login** imprime a mensagem do dia por meio de **/etc/motd** e executa um shell.
- O shell executa os arquivos de inicialização apropriados.[8]
- O shell imprime um prompt e aguarda entrada por parte do usuário.

Quando o usuário sai do sistema, o controle retorna para **init**, que desperta e gera um novo **getty** na porta do terminal.

A maior parte da configurabilidade nessa cadeia de eventos se concentra em **/etc/inittab**, onde o complemento normal do sistema de **getty**s é definido, e em **/etc/gettydefs**, onde algumas versões de **getty** procuram informações adicionais de configuração.

---

7. Se as senhas shadow não estiverem sendo usadas, a senha poderá provir diretamente de **/etc/passwd**. Além disso, **/etc/passwd** poderia ser suplantado ou complementado por um sistema de banco de dados administrativo como o NIS. Veja o Capítulo 18 para obter maiores informações.
8. **.profile** para **sh** e **ksh**; **.bash_profile** e **.bashrc** para **bash**; **.cshr** e **.login** para **csh/tcsh**.

Snow666

O arquivo /etc/inittab

init suporta vários "níveis de execução" que determinam quais recursos de sistema são habilitados. Há sete níveis de execução numerados de 0 a 6, com "S" reconhecido como sinônimo para o nível 1 (operação monousuário). Ao deixar o modo monousuário, **init** lhe solicita para entrar num nível de execução, a menos que exista um campo `initdefault` em **/etc/inittab** conforme descrito abaixo. **init** varre então o arquivo **inittab** em busca de todas as linhas que atendem ao nível de execução especificado.

Os níveis de execução são normalmente configurados para que tenhamos um nível no qual somente a console é habilitada e um outro nível que habilita todos os **getty**s. Podemos definir os níveis de execução da maneira que acharmos apropriada para o nosso sistema; entretanto, recomendamos que você não se afaste muito dos padrões.

Entradas em **inittab** se apresentam na forma

`id:run-levels:action:process`

Eis alguns exemplos simples de entradas **inittab**.

```
# Captura CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# Executa gettys em níveis de execução padrão
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
```

Nesse formato, *`id`* é uma string de um ou dois caracteres usada para identificar a entrada; ela pode ser nula. Para entradas de terminais, é costumeiro usar-se o número do terminal como *`id`*.

`run-levels` enumera os níveis de execução ao qual a entrada pertence. Se não for especificado nenhum nível de execução (como acontece na primeira linha), então a entrada é válida para todos os níveis de execução. *`action`* informa como manipular o campo *`process`*; a Tabela 7.6 lista alguns dos valores mais comumente usados.

Tabela 7.6 Valores comuns para o campo *action* de /etc/inittab.

Valor	Aguardar?	Significado
`initdefault`	–	Configura o nível de execução inicial
`boot`	Não	Executado quando **inittab** é lido pela primeira vez
`bootwait`	Sim	Executado quando **inittab** é lido pela primeira vez
ctrlaltdel	Não	Executado em resposta ao acionamento das teclas <Control-Alt-Delete>
`once`	Não	Inicia o processo de uma vez só
`wait`	Sim	Inicia o processo de uma vez só
`respawn`	Não	Sempre mantém o processo em execução
`powerfail`	Não	Executado quando **init** recebe um sinal de falha na alimentação
`powerwait`	Sim	Executado quando **init** recebe um sinal de falha na alimentação
`sysinit`	Sim	Executado antes de se acessar a console

Se um dos *`run-levels`* coincidir com o nível de execução atual e o campo *`action`* indicar que a entrada é relevante, **init** usará **sh** para executar (ou terminar) o comando especificado no campo *`process`*. A coluna Aguardar? da Tabela 7.6 informa se *`init`* aguardará pelo término do comando antes de prosseguir.

Nas linhas **inittab** do exemplo acima, as duas últimas linhas geram processos **mingetty** nas primeiras duas consoles virtuais (acessado com <Alt-F1> e <Alt-F2>). Se você adicionar terminais hardwired ou modems de discagem, as linhas **inittab** apropriadas ficarão parecidas com essas. Porém, você tem de usar **mgetty** ou **getty** (**agetty** no SuSE) com tais dispositivos, pois **mingetty** não é sofisticado o bastante para lidar com eles corretamente. Em geral, `respawn` é a ação correta e 2345 é um conjunto de níveis apropriado.

O comando **telinit -q** faz com que **init** releia o arquivo **inittab**.

gettys diversos requerem procedimentos de configuração distintos. A versão **getty/agetty** encontrada em SuSE e Debian geralmente é um pouco mais organizada do que **mgetty**, pois ela aceita todas as informações de configuração na linha de comando (em **/etc/inittab**).

www.snow666.blogspot.com
Snow666

O modelo genérico é

```
/sbin/getty port speed termtype
```

onde *port* é o arquivo de dispositivos da porta serial relativa a **/dev**, *speed* é a taxa de transmissão (por exemplo, 38400) e *termtype* identifica o tipo de terminal padrão para a porta. *termtype* se refere a uma entrada no banco de dados **terminfo**. A maioria dos emuladores simula um terminal DEC VT100, denotado por **vt100**. Há muitas outras opções secundárias, a maioria das quais se relaciona com a manipulação de modems de recepção de conexão (dial-in).

*Veja maiores informações sobre o banco de dados **terminfo**, mais adiante nesta seção.*

mgetty fornecido pelo Red Hat e SuSE é um pouco mais sofisticado do que **agetty** em seu tratamento com os modems e integra uma capacidade tanto para receber como para transmitir faxes. Infelizmente, sua configuração é um pouco mais difusa. Além de outros flags de linha de comando, **mgetty** pode aceitar uma referência opcional a uma entrada em **/etc/gettydefs** que especifica detalhes de configuração para o driver serial A menos que estejamos configurando uma configuração de modem sofisticada, normalmente podemos nos dar bem sem uma entrada **gettydefs**.

Em sistemas Red Hat, use **man mgettydefs** para localizar a página de manual para o arquivo **gettydefs**. Ele é chamado assim para evitar conflitos com uma página de manual **gettydefs** mais antiga que já não existe mais em nenhum sistema Linux.

Uma linha de comando **mgetty** simples para um terminal hardwired se parece com o seguinte:

```
/sbin/mgetty -rs speed device
```

speed é a taxa de transmissão (por exemplo, 38400) e *device* é o arquivo de dispositivos para a porta serial (use o nome de caminho completo).

Caso queira especificar um tipo de terminal padrão para uma porta ao usar **mgetty**, você tem de fazer assim num arquivo distinto, **/etc/ttytype**, e não na linha de comando **mgetty**. O formato de uma entrada em **ttytype** é:

```
termtype device
```

device é o nome abreviado para o arquivo de dispositivos representando a porta e *termtype* é o nome da entrada **terminfo** apropriada (veja a próxima seção). Por exemplo:

```
linux    tty1
linux    tty2
vt100    ttyS0
vt100    ttyS1
```

Suporte a terminais: os bancos de dados termcap e terminfo

O Linux suporta vários tipos de terminais diferentes pelo emprego de um banco de dados de capacidades de terminal que especifica os recursos e as peculiaridades de cada tipo de terminal. Há, historicamente, dois tipos de formatos de bancos de dados de terminal concorrentes: **termcap** e **terminfo**. Para maior compatibilidade, as distribuições do Linux geralmente fornecem ambos. O banco de dados **termcap** é contido no arquivo **/etc/termcap** e o banco de dados **terminfo** é armazenado sob **/usr/share/terminfo**. Os dois bancos de dados são similares e usam tipicamente o mesmo nome para cada tipo de terminal, de modo que a distinção entre eles não é importante.

Na forma como vêm, ambos os bancos de dados contêm entradas para centenas de terminais diferentes. Nesta era em que os terminais estão em extinção, a maioria deles é irrelevante. Uma boa regra prática é que tudo emula um terminal DEC VT100 até provem o contrário. Muitos emuladores também suportam "ansi"; "linux" e "xterm" são úteis para, respectivamente, consoles Linux e janelas **xterm** (terminal X Windows).

Veja a seção 7.12 para obter maiores informações sobre como configurar terminais no momento do login.

Os programas Linux examinam a variável de ambiente TERM para determinar que tipo de terminal estão usando. O terminal pode então ser pesquisado em **termcap** ou **terminfo**. Normalmente o sistema configura a variável TERM para nós no momento do login, de acordo com os argumentos da linha de comandos para **getty/agetty** ou com o conteúdo de **/etc/ttytype**.

Snow666

7.10 Caracteres Especiais e o Driver de Terminal

O driver de terminal suporta várias funções especiais que podemos acessar digitando determinadas teclas (normalmente teclas Control). A associação exata de funções às teclas pode ser configurada através dos comandos **tset** e **stty**. A Tabela 7.7 lista algumas dessas funções, juntamente com suas associações de teclas básicas.

Tabela 7.7 Caracteres especiais para o driver de terminal.

Nome	Padrão	Função
erase	<Control-?>	Apaga um caractere da entrada
werase	<Control-W>	Apaga uma palavra da entrada
kill	<Control-U>	Apaga a linha inteira da entrada
eof	<Control-D>	Envia uma indicação "EOF" (final de arquivo)
intr	<Control-C>	Interrompe o processo que está em execução no momento
quit	<Control-\>	Encerra o processo atual com um core dump
stop	<Control-S>	Pára a saída da tela
start	<Control-Q>	Reinicia a saída para a tela
susp	<Control-Z>	Suspende o processo atual
lnext	<Control-V>	Interpreta o próximo caractere literalmente

Por padrão, as versões PC do kernel Linux geram um caractere Delete (<Control-?>) quando a tecla Backspace é pressionada. (Essa tecla pode estar identificada como "backspace" ou "delete" ou poderá então apresentar apenas o desenho de uma seta invertida. Depende do teclado.) No passado, muitos sistemas UNIX usavam o caractere backspace (<Control-H>) para essa função. Infelizmente, a existência de dois padrões para a função cria uma série de problemas.

Podemos usar **stty erase** (veja abaixo) para informar ao driver de terminal qual código de tecla nossa configuração está realmente gerando. Entretanto, alguns programas (como os editores de texto e shells com recursos de edição de comandos) possuem uma concepção própria do que representa o caractere backspace e nem sempre eles prestam atenção à configuração do driver de terminal. Num truque útil, porém confuso, alguns programas obedecem tanto ao caractere backspace como ao caractere delete. Você também poderá descobrir que sistemas remotos nos quais você faz o login através da rede fazem suposições completamente diferentes daquelas do seu sistema local.

Solucionar esses pequenos conflitos pode ser por si só uma empreitada para um domingão. Geralmente não há nenhuma solução simples e universal. Cada pacote de software deve ser avaliado individualmente. Dois recursos úteis para ajudá-lo nessa tarefa são o *Linux Backspace/Delete mini-HOWTO* no site www.linux.org e um artigo formidável de Anne Baretta em www.ibb.net/~anne/keyboard.html.

7.11 Stty: Opções de Configuração de Terminais

stty permite que alteremos diretamente e consultemos as diversas configurações do driver de terminal. Há um trilhão de opções, porém a maioria delas pode ser certamente ignorada. **stty** geralmente usa os mesmos termos para opções de drivers que aqueles usados pela página de manual **termios**, porém há algumas discrepâncias ocasionais.

As opções de linha de comando do **stty** podem aparecer em qualquer ordem e em qualquer combinação. Um traço antes de uma opção, a nega. Por exemplo, para configurar um terminal para operação em 9600 bps com paridade par e sem tabulações de hardware, use o seguinte comando:

```
% stty 9600 even -tabs
```

Uma boa combinação de opções para ser usada num terminal típico seria

```
% stty intr ^C kill ^U erase ^? -tabs
```

Aqui, **-tabs** impede que o driver de terminal tire proveito do mecanismo de tabulação embutido do terminal, o que é útil, pois vários emuladores não são muito inteligentes quanto às tabulações. As demais opções configuram os caracteres de interromper, terminar e apagar associados, respectivamente, a <Control-C>, <Control-U> e a <Control-?> (delete).

Podemos usar **stty** para examinar os modos atuais do driver de terminal, bem como para configurá-los. **stty** sem nenhum argumento gera uma saída como a mostrada a seguir:

Snow666


```
% stty
speed 38400 baud; line = 0;
-brkint -imaxbel
```

Para um relatório de estado mais prolixo, use a opção **-a**:

```
% stty -a
speed 38400 baud; rows 50; columns 80; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 -hupcl -cstopb cread -clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke
```

O formato da saída é similar, porém apresenta um número maior de informações. O significado da saída deveria ser intuitivamente óbvio (caso você tenha escrito um driver de terminal recentemente).

stty opera no descritor de arquivos de sua entrada-padrão, de forma que é possível configurarmos e consultarmos os modos de um terminal que não sejam aquele atual usando o caractere de redirecionamento de entrada do shell (<). Você tem de ser o superusuário para alterar os modos de terminais de outras pessoas.

7.12 TSET: Configura Opções Automaticamente

tset inicializa o driver de terminal para um modo apropriado para um determinado tipo de terminal. O tipo pode ser especificado na linha de comandos; se ele não for especificado, **tset** usará o valor da variável de ambiente TERM.

tset suporta uma sintaxe para associação de valores da variável de ambiente TERM a outros valores. Esse recurso é útil caso você entre freqüentemente no sistema via modem ou através de comutador de dados e quiser ter o driver de terminal configurado corretamente para o terminal que você está realmente usando na outra ponta da conexão em vez de algo genérico e inútil como "discado".

Suponhamos, por exemplo, que usemos **xterm** em casa e que o sistema para o qual estamos discando seja configurado para pensar que o tipo de terminal de um modem é "discado". Se colocarmos o comando

```
tset -m dialup:xterm
```

no arquivo **.login** ou **.profile**, o driver de terminal será configurado apropriadamente para **xterm** independentemente de discarmos ou não.

Infelizmente, o comando **tset** não é realmente tão simples quanto parece ser. Para fazermos com que **tset** ajuste nossas variáveis de ambiente além de configurar os modos de terminal, precisaremos de algo como o seguinte:

```
set noglob
eval `tset -s -Q -m dialup:xterm`
unset noglob
```

Essa mágica suprime as mensagens que **tset** normalmente envia (o flag **-Q**) e solicita que os comandos de shell configurem para que as variáveis de ambiente sejam geradas como saída em seu lugar (o flag **-s**). Os comandos de shell emitidos por **tset** são capturados pelas aspas simples invertidas e alimentados para o shell como entrada por meio do comando embutido **eval** fazendo com que os comandos tenham o mesmo efeito similar a estarem sendo digitados pelo usuário.

set noglob impede que o shell expanda quaisquer metacaracteres, como "*" e "?", que sejam incluídos na saída de **tset**. Esse comando não é necessário para usuários **sh/ksh** (nem o comando **unset noglob** para desfazê-lo), já que esses shells normalmente não expandem caracteres entre aspas simples invertidas. O comando **tset** é o mesmo independentemente de o shell usado; **tset** examina a variável de ambiente SHELL para determinar que tipo de comandos deve apresentar.

7.13 Como Desfazer a Intrusão Num Terminal

Alguns programas (como o **vi**, por exemplo) promovem alterações drásticas no estado do driver de terminal enquanto estão sendo executados. Essa intervenção normalmente é invisível para o usuário, já que o estado de terminal é cuidadosamente restaurado toda vez que o programa é fechado ou suspenso. Porém, é possível que um programa trave

www.snow666.blogspot.com
Snow666

ou seja interrompido sem executar suas tarefas de fechamento normal. Quando isso acontece, um terminal pode se comportar de maneira muito estranha: ele pode falhar ao tratar linhas novas corretamente, ao ecoar caracteres que digitamos ou ao executar comandos de forma apropriada.

Uma outra maneira comum de confundir um terminal é executar acidentalmente **cat** ou **more** em um arquivo binário. A maioria dos arquivos binários contém uma mistura de caracteres especiais capazes de certamente mandar para o espaço emuladores menos robustos.

Para corrigir essa situação, podemos usar **reset** ou **stty sane**. **reset** é, na verdade, um link para **tset** e ele é capaz de aceitar a maioria dos argumentos de **tset**. Entretanto, ele normalmente é executado sem argumentos. Tanto **reset** quanto **stty sane** restauram o funcionamento correto do driver de terminal e envian um código de reinicialização apropriado por meio de **termcap/terminfo**, caso um deles esteja disponível.

Em muitos casos em que um **reset** é apropriado, o terminal foi deixado num modo no qual nenhum processamento estava sendo feito nos caracteres digitados pelo usuário. A maioria dos terminais gera carriage returns em vez de caracteres de linha nova quando a tecla Return ou Enter é pressionada; sem processamento de entrada, essa tecla gera caracteres <Control-M> em vez de enviar o comando atual para ser executado. Para introduzir diretamente caracteres de linha nova, use <Control-J> ou a tecla de linefeed (caso exista uma) em vez da tecla Return.

7.14 Modems

Um modem converte o sinal serial digital produzido por um computador num sinal análogo compatível para transmissão numa linha telefônica padrão. Os modems são usados para uma série de aplicações. Veja a página 218 para um exemplo típico.

Modems externos possuem um conector RJ-11 no lado análogo e uma interface RS-232 de algum tipo no lado digital — usualmente uma fêmea DB-25. Na parte frontal eles normalmente têm uma série de leds que exibem o estado atual e nível de atividade do modem. Esses leds são incrivelmente úteis para depuração, de modo que os modems devem geralmente ser localizados em algum ponto de fácil visão.

Modems internos são normalmente vistos somente em PCs. Eles são encaixados num slot ISA, PCI ou PCM-CIA e possuem um conector RJ-11 que fica na parte externa traseira do gabinete do computador uma vez que o modem é instalado. Eles são mais baratos do que um modem externo, porém mais problemáticos para serem configurados e geralmente não possuem os leds indicadores.

Caso esteja considerando empregar um modem interno, terá de verificar para ter certeza de que é suportado pelo Linux. CPUs rápidas tornaram possível simplificar o hardware dos modems, executando algumas tarefas de processamento de sinais no processador do computador. Infelizmente, modems que operam dessa maneira (conhecidos genericamente como Winmodems) requerem drivers sofisticados e não são suportados universalmente no ambiente Linux. Consulte *Linmodem HOWTO* de Sean Walbran e Marvin Stodolsky (disponível em www.linux.org) para ter uma visão geral do suporte a Winmodems no ambiente Linux. (*Modem HOWTO* também é muito útil para uma perspectiva mais ampla sobre o gerenciamento de modems.)

Modems internos são normalmente feitos para parecer como se estivessem conectados através de uma porta serial fantasma sob a perspectiva do software em nível de usuário. Essa convenção ajuda a isolar a função lógica do modem de sua implementação de hardware. Pacotes de software padrão são capazes de acionar o modem sem ter de saber nada sobre suas peculiaridades.

Os modems variam um tanto em termos de robustez genérica, porém esta característica é difícil de ser avaliada sem experiência direta. No passado, constatamos que alguns modems eram significativamente mais tolerantes em termos de ruído de linha do que outros. Hoje, a maioria dos projetos usa um chipset-padrão de um dos vários grandes fabricantes, de maneira que provavelmente as variações entre os modems não são tão grandes quanto foram no passado.

Os modems de alta velocidade requerem firmware complexo e este firmware eventualmente apresenta bugs. Os fabricantes compartilham firmware entre modelos quando possível, de modo que firmware bom ou ruim apresentam uma tendência de serem usados em linhas de produtos. Por essa razão, ainda recomendamos utilizar as marcas bem conhecidas.

Modulação, correção de erros e protocolos de compactação de dados

Há muito tempo, era importante verificar os protocolos exatamente suportados por um modem, pois os padrões estavam em constante mudança e os fabricantes de modems nem sempre implementavam um conjunto de protocolos completo. Atualmente, todos os modems suportam praticamente os mesmos padrões. A única diferença real entre eles está na qualidade do firmware, na eletrônica e no suporte.

www.snow666.blogspot.com
Snow666

A taxa de transmissão de um protocolo é a taxa na qual o sinal de portadora é modulado. Se houver mais de dois níveis de sinal, mais do que um bloco de informações pode ser enviado por transição e a velocidade em bits por segundo será maior do que a taxa de transmissão. Historicamente, a velocidade de dados e a velocidade de sinalização dos modems eram as mesmas, induzindo a uma casual confusão dos termos "baud" e "bps" (bits por segundo).

A maioria dos modems atual usa o padrão V.90 "56K", que não significa que ele forneça realmente uma capacidade de 56 Kb/s. Na melhor das hipóteses, ele possibilita uma taxa de 33,6 Kb/s do computador para o ISP (provedor de serviços Internet) e de 53 Kb/s na direção oposta. Não obstante, o padrão V.90 atinge velocidades muito próximas dos limites teóricos e legais de sinalização por meio de linhas telefônicas de voz comuns e não se espera que sejam superados em breve. [9]

O ruído de linha pode introduzir um número de erros significativos numa conexão via modem. Diversos protocolos de correção de erros foram desenvolvidos para empacotar os dados transmitidos e fornecer correção de erros baseados em técnicas de checksum, isolando o usuário ou aplicativo das falhas da linha. Costumava ser necessário ter algum conhecimento sobre isso para poder configurar um modem corretamente, porém nos dias de hoje ele simplesmente funciona.

Os algoritmos de compactação de dados podem ser usados para dividir o número de bits que devem ser transmitidos entre extremidades analógicas. O nível de compactação varia de pior do que nada (quando se transmite dados que já foram compactados) até uma razão de 4:1. Um valor mais típico é de 1,5:1. Em geral, a configuração mediana apresenta um melhor desempenho quando algum desses algoritmos de compactação é ativado.

minicom: dial out

Os programas dial-out UNIX tradicionais **tip** e **cu** são relativamente impopulares nos sistemas Linux embora ambos tenham sido portados (**cu** é normalmente empacotado com UUCP, um sistema de comunicação telefônica obsoleto). Mais comum sob o Linux são pacotes abrangentes como **kermit** e **minicom** que fornecem emulação de terminais e suporte para protocolos de transferência de dados. Para depuração e uso ocasional, recomendamos o **minicom**, em grande parte porque é o que tem maior possibilidade de já estar pré-instalado. O Debian não o instala como padrão; execute **apt-get install minicom** caso pareça que ele está faltando.

Para melhor ou para pior, **minicom** está mais para software tipo PC do que para Linux. Se suas memórias lá pelos idos de 1986 incluíam logar a 1200 bauds através de um emulador de terminal DOS enquanto descolava algumas notas de cem para comprar cocaína, **minicom** lhe fará sentir-se em casa. Para configurar o software, execute **minicom -s** como root e entre no menu "serial port setup". Configure o arquivo de dispositivos para o modem, ative o controle de fluxo via hardware, configure a codificação para 8N1 (8 bits de dados, paridade ímpar e 1 stop bit) e certifique-se de que a velocidade esteja OK. Retorne para o menu principal e selecione "save settings as dfl" para gravar suas alterações.

Caso esteja familiarizado com a linguagem de comandos Hayes usada pela maioria dos modems, poderá simplesmente introduzir os comandos diretamente (por exemplo, "ATDT5551212" para discar 555-1212). Para que o **minicom** faça isso para você, digite <Control-A> e D para entrar no menu de discagem.

Modems bidirecionais

Normalmente é prático usar um único modem tanto para serviços de discagem de entrada como de saída, particularmente se quiser usar o suporte a fax. Essa configuração requer que **getty** manipule a porta serial de uma maneira especialmente solícita, já que ele não pode simplesmente cuidar da porta e bloquear todos os demais processos. O mecanismo exato pelo qual o compartilhamento é atingido é um tanto sutil e envolve tanto recursos de driver serial quanto arquivos criados sob **/var/lock** para sinalizar as intenções dos diversos participantes. Todos os programas que compartilham a porta devem obedecer ao protocolo apropriado.

No passado, configurar um modem para uso bidirecional era um grande problema que exigia muitos ajustes de sistema específicos e normalmente não funcionava muito bem. Felizmente, os principais pacotes de software Linux se dão relativamente bem uns com os outros da forma como vêm configurados originalmente desde que você use **mgetty** ou **uugetty**. Em geral, prefere-se **mgetty**; é o comportamento-padrão para compartilhamento, de modo que basta você ligar e sair rodando.

7.15 Depurando uma Linha Serial

Depurar linhas seriais não é difícil. Alguns erros típicos são:

9. Um crítico escreveu: "Não era isso que eles diziam em relação aos 110 bauds?".

Snow666

- Esquecer-se de informar a **init** para reler seus arquivos de configuração.
- Esquecer-se de configurar a portadora via software ao usar cabos de três fios.
- Usar um cabo com a nulidade incorreta.
- Soldar ou grampear conectores DB-25 de ponta-cabeça.
- Conectar um dispositivo ao fio errado devido a associações de fiação incorretas ou inexistentes.
- Configurar as opções de terminal incorretamente.

Um dispositivo de interconexão é uma ferramenta indispensável para depurar problemas de cabeamento. Ele é conectado na linha serial e exibe os sinais em cada pino à medida que passam através do cabo. Os melhores dispositivos desse tipo possuem tanto conectores macho quanto fêmea em cada lado e são totalmente flexíveis e bissexuais em seus posicionamentos. LEDs associados a cada pino "interessante" (pinos 2, 3, 4, 5, 6, 8 e 20) mostram quando o pino está ativo.

Alguns dispositivos de interconexão são somente para leitura e apenas permitem o monitoramento de sinais; outros permitem que refaçamos a conexão e garantamos uma determinada tensão num dado pino. Se você suspeita, por exemplo, que um cabo precisasse ser anulado (cruzado), poderíamos usar o dispositivo de interconexão para prevalecer em relação à verdadeira fiação e inverter os pinos 2 e 3 e também os pinos 6 e 20.

Veja a seção 15.14, Capítulo 15, para obter maiores informações sobre a Black Box.

Um dispositivo de interconexão ruim pode ser pior do que simplesmente não usar nenhum dispositivo de interconexão. Nossa implementação favorita é a BOB-CAT-B produzida pela Black Box. É um dispositivo fácil de usar que custa cerca de US$ 250. Para contactar a Black Box: (724)746-5500 ou www.blackbox.com.

7.16 Outras Portas de E/S Comuns

As portas seriais costumavam ser o padrão simples para conexão de periféricos de baixa velocidade a sistemas UNIX, porém a plataforma de hardware baseada em PCs abriu uma série de opções novas e interessantes.

As portas paralelas estilo PC são similares em conceito às portas seriais, porém elas transferem oito bits de dados de uma só vez em vez de 1 bit. Elas são significativamente mais rápidas do que as portas seriais, mas exigem conectores e cabeamento mais volumosos. As interfaces paralelas são encontradas mais comumente em impressoras, mas no mundo Windows elas também são usadas para conectar Zip drives e unidades de fita que requerem uma largura de banda maior do que uma porta serial é capaz de oferecer. Porém, o suporte Linux para dispositivos paralelos que não sejam impressoras é escasso. Muitos dispositivos que utilizam a porta paralela para conexão, como Zip drives, scanners etc, são suportados no Linux através de uma emulação de conexão SCSI feita no kernel. Várias distribuições incluem esse suporte nos kernels-padrão distribuídos, de modo que facilite o suporte a esses dispositivos. Consulte *Zip-Drive Mini-HOWTO* de *Kyle Dansie* para mais informações (*disponível em www.linux.org*).

O USB (Universal Serial Bus, barramento serial universal) é uma inovação mais recente que coloca portas paralelas e seriais tradicionais no chinelo. Ele é mais rápido (até 12 Mb/s), arquitetonicamente elegante e emprega cabos padronizados que são ao mesmo tempo baratos e simples. Infelizmente, levará anos para as empresas se livrarem de seus dispositivos seriais e paralelos existentes. Por enquanto, podemos somente sonhar em viver na terra prometida do USB (e esperar que a Microsoft não chegue e construa em todos os bons lotes primeiro).

Portas paralelas

Qualquer sistema Linux em PC terá uma porta paralela, porém alguns fabricantes também começaram a adicionar portas paralelas em estações de trabalho dedicadas de modo que os "maníacos" por Linux são capazes de usar impressoras feitas para o mercado Windows.

As portas paralelas aderiram a cinco ou seis padrões de protocolos distintos ao longo dos anos, porém interfaces paralelas contemporâneas são compatíveis com todas as versões anteriores. O melhor padrão disponível atualmente é o IEEE-1284 que incorpora compatibilidade com a maioria dos padrões importantes (tanto de fato como escrito).

Para alcançar as velocidades mais altas, as portas paralelas podem ser configuradas para operarem tanto no modo EPP (Enhanced Parallel Port) quando no modo ECP (Enhanced Capability Port), possibilitando velocidades de 2 MB/s ou mais. Os dois modos de alta velocidade são mais ou menos equivalentes, exceto pelo fato de o ECP suportar DMA. É pouco provável que esse recurso faça alguma diferença na prática.

Os computadores normalmente fornecem um conector DB-25 para a porta paralela e os periféricos tendem a ter um conector Centronics fêmea de 36 pinos. Um terceiro tipo de conector, mini-Centronics, também é permitido pelo IEEE-1284.

Snow666

Cabos paralelos podem ser estendidos até 10 metros. Já que os comprimentos de cabos são limitados e somente dois tipos de conectores são usados comumente, é mais eficiente em termos de custos comprar cabos paralelos pré-fabricados do que fazê-los por conta própria.

Embora muitos periféricos Windows sejam capazes de se conectarem a uma porta paralela de um PC, somente impressoras são ampla e genericamente suportadas no Linux. Outros periféricos, como Zip drives e câmeras de vídeo, precisam de um driver específico para dispositivo para serem utilizáveis. Os drivers estão disponíveis para uma série de dispositivos populares, porém os drivers têm de ser normalmente instalados manualmente. Veja a home page sobre portas paralelas no Linux em www.torque.net/linux-pp.html.
para obter maiores detalhes. Para uma lista genérica de FAQs sobre portas paralelas e informações relacionadas com portas paralelas, pesquise "parallel port" em www.webopedia.com.

USB: o Barramento Serial Universal

O USB é um sistema de interconexão de periféricos genérico desenvolvido pela Compaq, DEC, IBM, Intel, Microsoft, NEC e Northern Telecom. O primeiro padrão USB foi publicado em 1996. A aceitação do USB no mundo Windows rapidamente foi se avolumando como uma bola de neve ao longo dos últimos anos. Todos os novos PCs possuem portas USB e a maioria dos periféricos de computadores está disponível nas versões USB.

Para obter maiores informações sobre USB, veja o site www.usb.org.

O USB é um grande sistema e acreditamos que provavelmente ele permaneça em uso por muitos anos. Ele praticamente tem todas as propriedades e recursos que alguém poderia imaginar para um barramento de baixa velocidade.

- É extremamente barato.
- Podem ser conectados até 127 dispositivos.
- Os cabos possuem somente quatro fios: alimentação, terra e dois fios de sinal.
- Os conectores e os gêneros de conectores são padronizados.
- Os conectores são pequenos e os cabos são finos e flexíveis.
- Os dispositivos podem ser conectados e desconectados sem a necessidade de desligar o sistema.
- Pode-se atingir velocidades de sinalização de até 12 Mb/s.
- Dispositivos paralelos e seriais antigos podem ser conectados via adaptadores.

O USB pode até mesmo ser usado com tecnologia LAN, embora não seja realmente projetado para isto.

O Linux já possui um suporte relativamente amplo para dispositivos USB, embora esta ainda seja uma área de desenvolvimento ativa. Normalmente é necessário algum ajuste de configuração para que determinado dispositivo USB seja acionado e fique operacional; o grau de ajuste necessário varia conforme o tipo de dispositivo. O padrão USB define interfaces-padrão para várias classes de equipamentos comuns (como mouses, modems e dispositivos de armazenamento de massa), de forma que o emprego destes dispositivos normalmente é relativamente direto. Por outro lado, dispositivos como câmeras e scanners podem requerer drivers de chipset específicos.

Podemos verificar se um sistema tem um USB plenamente habilitado (inclusive suporte para plugamento com o sistema no ar) pesquisando um sistema de arquivos montado do tipo usbdevfs; o ponto de montagem normalmente será **/proc/bus/usb**. Todos os nossos sistemas de exemplo, exceto o Debian, são configurados por padrão. Caso não haja nenhum sistema de arquivos USB montado, talvez seja necessário reconfigurar o kernel ou instalar os scripts de plugamento com o sistema no ar (ou ambos). Veja as instruções em www.linux-usb.org para obter maiores detalhes.

O sistema de arquivos **/proc/bus/usb** fornece um instantâneo dinamicamente atualizado do barramento USB. Arquivos numerados (como **001**, que representa o controlador USB do próprio host) correspondem a dispositivos USB individuais que foram conectados ao sistema. O arquivo **/proc/bus/usb/devices** fornece informações detalhadas sobre o censo atual dos dispositivos, e o arquivo **/proc/bus/usb/drivers** contém os nomes dos drivers USB registrados atualmente (independentemente de eles estarem sendo usados ou não). Ambos os arquivos podem ser inspecionados com **more** ou seu editor de textos predileto.

Embora o sistema de arquivos **/proc/bus/usb** seja necessário para suportar a autoconfiguração e seja útil para depuração, ele em geral não é usado diretamente pelos drivers em nível de kernel. A maioria dos dispositivos USB é acessada por arquivos de dispositivos estilo UNIX tradicional sob **/dev**.

Para informações atualizadas sobre dispositivos suportados e os drivers necessários para fazê-los funcionar, consulte a lista de dispositivos em www.linux-usb.org.

Snow666

7.17 Exercícios

E7.1 O que é um cabo modem nulo? Como ele é usado ao conectar dispositivos seriais DCE e DTE?

E7.2 Podemos usar um cabo serial de 3 fios para uma conexão de modem serial? Para uma impressora serial? Por que ou por que não?

E7.3 Como funciona o controle de fluxo de hardware? O que pode ser feito caso um sistema não entenda o controle de fluxo de hardware?

E7.4 O que é um pseudoterminal? Quais programas usam pseudoterminais?

E7.5 Forneça entradas **inittab** que

a) Rodem um programa chamado **server-fallback**, aguarde sua finalização e depois pare imediatamente o sistema caso falhe a alimentação.

b) Gere novamente um servidor chamado **unstable-srv** caso ele trave.

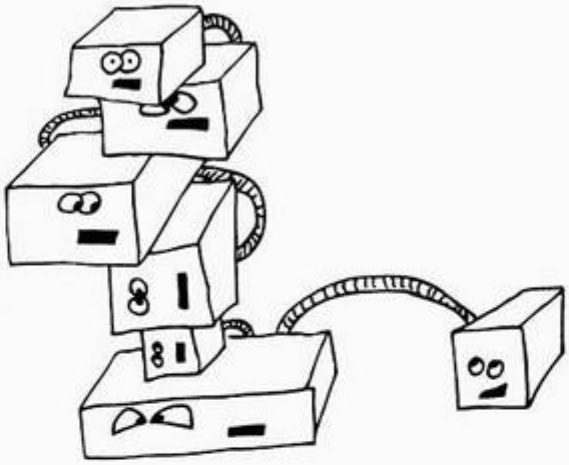
c) Execute um script chamado **clean-temp** que elimine todos os arquivos temporários cada vez que o sistema é inicializado.

★ **E7.6** Compare os padrões serial RS-232 e USB.

★ **E7.7** Um amigo descuidadamente foi embora e deixou seu terminal logado a madrugada toda no laboratório Linux e agora o laboratório está passando por problemas estranhos ao executar aplicativos de shell. Os programas sumiram, e entradas anteriores somem quando certos comandos e entradas são dados; entretanto, algumas coisas parecem funcionar normalmente. O que um usuário inimigo poderia ter feito para provocar tal comportamento? Explique como você poderia testar sua resposta. Como o problema poderia ser corrigido? Quem poderia ter dado tal golpe baixo?

www.snow666.blogspot.com
Snow666

8 Adicionando um Disco



É difícil de acreditar que com os avanços das últimas décadas em termos de chip, rede e de tecnologia de software, ainda usemos basicamente a mesma tecnologia para armazenamento de dados a longo prazo que já era popular há 40 anos. As capacidades aumentaram (e os preços caíram) em grandes fatores, porém os conceitos básicos permanecem inalterados.

Infelizmente, novos empregos para consumir espaço em disco vêm surgindo a cada dia, especialmente quando as redes de longa distância se tornaram comuns. Como resultado, ainda há aquele velho problema de nunca termos espaço em disco suficiente. No momento em que um novo disco é adicionado ao sistema, ele já se encontra com metade de sua capacidade esgotada; ou assim parece. Fazer com que os usuários limpem espaço em disco é tão difícil quanto fazer com que os adolescentes arrumem os seus quartos. Conseqüentemente, um administrador terá ocasionalmente de instalar novas unidades de disco.

A maioria dos sistemas de servidor se conecta por meio de um barramento periférico padrão chamado SCSI (Small Computer Systems Interface). Uma interface alternativa chamada IDE (Integrated Drive Electronics) é padrão nos PCs desktop e laptop. Começamos este capítulo com uma discussão geral sobre os padrões SCSI e IDE e a estrutura dos discos rígidos modernos. A seguir discutimos os mecanismos gerais pelos quais os discos são formados e particionados e o procedimento para inicializar sistemas de arquivos.

8.1 Interfaces de Disco

Hoje, somente alguns poucos padrões de interface são usados, embora surjam várias tecnologias novas. É importante selecionar unidades de disco que combinem com as interfaces do sistema nas quais elas serão instaladas. Se um sistema suporta várias interfaces diferentes, você deve usar aquela que melhor atenda às suas necessidades em termos de velocidade, redundância, mobilidade e preço:

- SCSI é uma das interfaces de disco mais comumente e amplamente suportadas. Ela vem em diversas variantes, todas as quais suportam discos múltiplos num barramento e várias velocidades e estilos de comunicação. SCSI é descrito com mais detalhes na próxima seção.
- O IDE foi desenvolvido como uma interface simples e de baixo custo para PCs. Originalmente ela foi chamada de Integrated Drive Electronics, pois colocava o controlador de hardware no mesmo dispositivo que as lâminas de disco e usava um protocolo de alto nível para comunicação entre o computador e os discos. Esta agora é a arquitetura-padrão para todos os discos modernos, porém o mesmo nome persiste. Os discos IDE apresentam uma velocidade de média para rápida, alta capacidade de armazenamento e são incrivelmente baratos. Entretanto, o desenho da interface torna o IDE uma opção prática somente para estações

Snow666

de trabalho com quatro ou menos dispositivos. Veja *A interface IDE*, mais adiante, nesta seção, para obter maiores informações sobre o IDE.

- Fibre Channel é uma interface serial que está ganhando popularidade no ambiente corporativo graças à sua alta largura de banda e ao grande número de dispositivos que podem ser conectados a ela de uma só vez. Os dispositivos Fibre Channel se conectam por meio de uma fibra ótica ou cabo de cobre axial duplo. As velocidades atuais são de 100 MB/s ou mais. Topologias comuns incluem loops, chamados FC-AL (Fibre Channel Arbitrated Loop), e estruturas que são construídas com switches Fibre Channel. Fibre Channel é capaz de "falar" vários protocolos diferentes, inclusive SCSI e até mesmo IP. Dispositivos Fibre Channel são identificados por um número de ID gravado internamente, chamado World Wide Name que é similar a um endereço MAC Ethernet.
- O USB (Universal Serial Bus, barramento serial universal) se tornou popular para conexão de dispositivos como teclados e mouses, porém ele possui largura de banda suficiente para suportar dispositivos de disco mais lentos como discos rígidos removíveis e unidades de CD-ROM. O USB é comum em PCs e nos permite facilmente transferir um disco entre sistemas.

SCSI e IDE são de longe os concorrentes dominantes no mercado de unidades de disco. Essas são as únicas interfaces que discutiremos detalhadamente.

A interface SCSI

Vários chipsets implementam o padrão SCSI, de modo que os fornecedores algumas vezes fornecem suporte SCSI diretamente na placa-mãe. SCSI define um canal de dados genérico que pode ser usado por todos os tipos de periféricos. Mais comumente, ele é usado para discos, unidades de fita, scanners e impressoras. O padrão SCSI não especifica como um disco é construído ou disposto, somente a maneira pela qual ele se comunica com outros dispositivos.

O padrão SCSI passou por diversas revisões, com o SCSI-3 sendo a versão atual. SCSI-1 foi desenvolvido em 1986 como um padrão ANSI baseado no SASI (Shugart Associates System Interface) que era um sistema de barramento comercialmente disponível. O SCSI-2 foi desenvolvido em 1990. Ele é compatível com a versão anterior SCSI-1, porém acrescenta diversos recursos em termos de desempenho. Entre esses recursos, temos enfileiramento de comandos, que permite aos dispositivos reordenar solicitações de I/O para otimizar a vazão "scatter-gather" I/O que permite DMA (Direct Memory Access, acesso direto à memória) em regiões de memória descontínuas.

Talvez você possa ver os termos "fast" e "wide" aplicados ao dispositivo SCSI-2; estes significam que a velocidade de barramento é dobrada ou que o número de bits transferidos simultaneamente é maior, tipicamente 16 ou 32 bits em vez dos usuais 8 bits. Cadeias wide SCSI também podem suportar até 16 dispositivos; narrow SCSI permitem somente 8. [1] Velocidade e largura são recursos distintos comumente usados juntos para aumentos sinérgicos.

SCSI-3 é, na verdade, uma família de padrões. Ele inclui especificações para várias mídias físicas, inclusive os barramentos paralelos tradicionais e mídia serial de alta velocidade como Fibre Channel e IEEE 1394 ("FireWire"). Ele também define os conjuntos de comandos SCSI e introduz melhorias para suporte à autoconfiguração de dispositivos, aplicações multimídia e novos tipos de dispositivos.

Embora a especificação SCSI-3 ainda não tenha sido finalizada, muitos de seus recursos já se consolidaram no mercado, normalmente sob o nome "Ultra SCSI". O SCSI-3 engloba o SCSI-2, de forma que um certo grau de compatibilidade se encontra disponível. Tenha em mente, entretanto, que colocar um dispositivo mais velho num barramento mais novo pode diminuir a velocidade do barramento inteiro. Ele também afeta o comprimento máximo do cabo.

Um outro recurso algumas vezes usado em sistemas SCSI é o "SCSI diferencial". No SCSI normal ("terminação simples"), um pino sim outro não são aterrados para ajudar a reduzir a interferência entre sinais. [2] Essa construção limita o comprimento do cabo a 6 metros para o SCSI-1 e 3 e 3 metros para o SCSI-2. Ultra SCSI reduz o comprimento total do barramento ainda mais, para 1,5 m. Sinalização diferencial coloca um sinal invertido próximo a cada pino em vez de um terra, tornando a voltagem líquida igual a zero e reduzindo significativamente o ruído.

A sinalização diferencial aumenta o limite de comprimento de cabo para 25 m no caso do SCSI-2 e para 12 m para o Ultra SCSI. O comprimento extra é um grande ganho quando todos os discos (ou bibliotecas de fitas) estiverem num gabinete externo distante. Porém, já que a sinalização diferencial é completamente incompatível com dispositivos não diferenciais, você *tem* de se certificar de que tenha um controlador, disco, cabo e terminadores

1. Barramentos SCSI de 32 bits não são muito comuns. Alguns podem requerer vários cabos, conhecidos como o cabo A e o cabo B.
2. A interferência é mais significativa num barramento elétrico paralelo. Ela não preocupa em transmissões seriais e por fibra ótica como o Fibre Channel, que possuem um comprimento máximo de cabo muito maior.

Snow666

diferenciais. Identifique-os bem para garantir que você acidentalmente não os misture com dispositivos de terminação simples, ou as coisas não funcionarão.

A Tabela 8.1 sintetiza as diferentes versões SCSI e suas larguras de banda de barramentos e comprimentos de cabos associados.

O comprimento máximo de cabo para Ultra SCSI e Ultra Wide SCSI de terminação simples depende do número de dispositivos em uso. Para 8 dispositivos, 1,5 m é o máximo; se forem usados somente 4 dispositivos, o barramento pode ser estendido a 3 metros. Ultra Wide SCSI suporta todos os 16 dispositivos somente em modo diferencial.

Tabela 8.1 A evolução do SCSI.

Versão	Freq.	Largura	Velocidade	Comprimento	Compr. Dif.
SCSI-1	5 MHz	8 bits	5 MB/s	6m	25m
SCSI-2	5 MHz	8 bits	5 MB/s	6m	25m
Fast SCSI-2	10 MHz	8 bits	10 MB/s	3m	25m
Fast/wide SCSI-2	10 MHz	16 bits	20 MB/s	3m	25m
Ultra SCSI	20 MHz	8 bits	20 MB/s	1.5m[a]	25m
Ultra Wide SCSI[b]	20 MHz	16 bits	40 MB/s	1.5m[a]	25m
Ultra2 Wide SCSI[b]	40 MHz	16 bits	80 MB/s	_c	25m (HVD)[d] 12m (LVD)
Ultra3 Wide SCSI[e]	80 MHz	16 bits	160 MB/s	_c	12m (LVD)

a. Varia; veja os comentários no texto abaixo.
b. Ultra Wide SCSI e Ultra2 Wide SCSI são algumas vezes chamadas, respectivamente, de Fast-20 wide SCSI e Fast-40 wide SCSI.
c. Estas versões do SCSI usam apenas sinalização diferencial.
d. HVD significa High Voltage Differential (diferencial de tensão elevado) e LVD, Low Voltage Differential (diferencial de tensão baixo). O HVD é usado para as primeiras versões SCSI e não é definido acima de Ultra2 SCSI.
e. Ultra Wide 3 SCSI é algumas vezes chamada de Ultra-160.

São usados diversos tipos de conectores para dispositivos SCSI. Eles variam dependendo da versão do padrão SCSI em uso e do tipo de conexão: internos ou externos. Os dispositivos narrow SCSI possuem 50 pinos e os dispositivos wide SCSI possuem 68 pinos. Os dispositivos internos aceitam tipicamente um conector de 50 pinos ou um conector mini micro de 68 pinos macho conectado a um flat cable. As unidades externas normalmente se conectam ao computador por meio de um conector mini micro de alta densidade de 50 ou 68 pinos. A Apple reduziu os 50 pinos para 25 pinos conectando todas as linhas de terra e adaptando o barramento a um conector DB-25.

Uma variante interessante que é especialmente útil para os arrays de unidades que podem ser trocadas com o sistema em funcionamento é o plugue SCA (Single Connector Attachment). Trata-se de um conector de 80 pinos que inclui conexões de barramento, alimentação e configuração SCSI, permitindo a um único conector suprir todas as necessidades da unidade.

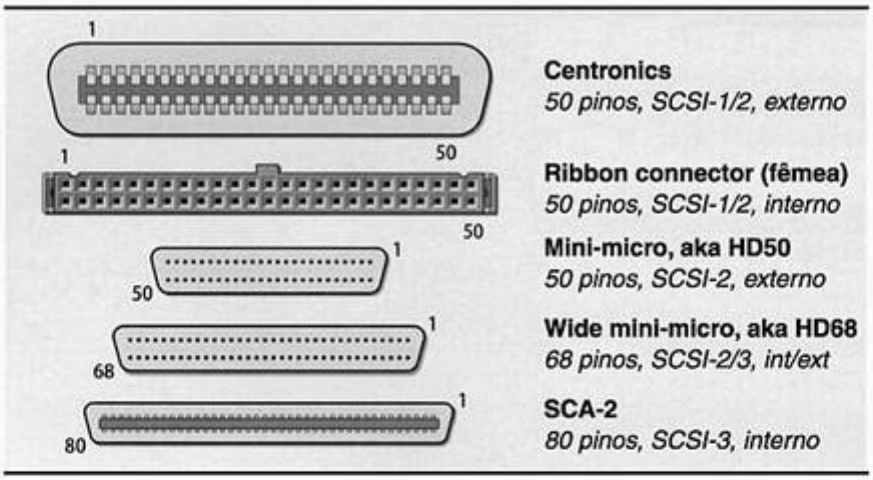
O Diagrama A da próxima página ilustra os conectores mais comuns. Cada conector é mostrado de frente, como se estivesse para ser plugado na sua parte frontal.

Os conectores em dispositivos SCSI são quase sempre fêmeas e as terminações de cabos SCSI são macho. Os barramentos SCSI usam uma configuração em cadeia de modo que a maioria dos dispositivos externos possui duas portas SCSI. As portas são idênticas e intercambiáveis, em que cada uma delas pode ser a entrada. Por alguma razão, os fabricantes de scanner parecem considerar-se isentos das regras normais da física e algumas vezes fornecem somente uma porta SCSI. Se ela não for terminada internamente, esses dispositivos exigirão um tipo especial de terminador.

Os dispositivos SCSI internos são usualmente conectados a um flat cable; somente uma porta é necessária no dispositivo SCSI em si pois os conectores podem ser conectados no meio do flat cable. Ao usar um flat cable, certifique-se de que o pino 1 do barramento SCSI esteja conectado ao pino 1 da unidade de disco rígido. (O pino 1 é normalmente marcado com uma faixa vermelha.)

Snow666

Diagrama A Conectores SCSI comuns (vista frontal, macho, exceto onde indicado em contrário)



Cada final do barramento SCSI deve ter um resistor de terminação ("terminador"). Esses resistores absorvem sinais à medida que eles atingem o final do barramento e evitam que ruído se reflita de volta para o barramento. Os terminadores assumem várias formas, desde pequenos plugues externos que ligamos numa porta regular a conjuntos de minúsculos pacotes de resistor que se instalam nas placas de circuito de um dispositivo. Alguns dispositivos são até mesmo autoterminados.

O SCSI-1 usava um desenho de terminador diferente daquele de versões SCSI anteriores ("passivo" em vez de "ativo"), de modo que terminadores antigos (e dispositivos autoterminados) podem provocar problemas num barramento SCSI mais novo. Na prática, raramente encontramos problemas com tais terminadores.

Uma extremidade do barramento normalmente termina dentro do computador host, seja num controlador SCSI como numa unidade SCSI interna. A outra ponta normalmente termina num dispositivo externo ou no controlador SCSI caso não haja dispositivos externos. Se você passar por problemas de hardware aparentemente aleatórios no seu barramento SCSI, verifique primeiramente se ambas as extremidades do barramento estão apropriadamente terminadas. Terminação imprópria é um dos enganos de configuração mais comuns em interfaces SCSI, e os erros por elas produzidos podem ser obscuros e intermitentes.

Cada dispositivo possui um endereço SCSI ou "número de destino" que o distingue dos demais dispositivos presentes no barramento. Os números de destino começam em 0 e vão até 7 ou 15, dependendo se o barramento é narrow ou wide. O controlador SCSI em si é contado como um dispositivo e normalmente é o destino 7 (mesmo num barramento wide, para fins de compatibilidade com versões anteriores). Todos os demais dispositivos devem ter seus números de destino configurados a valores únicos. É um erro comum se esquecer de que o controlador SCSI possui um número de destino e configurar um dispositivo para o mesmo número de destino do controlador.

Um endereço SCSI é basicamente arbitrário. Tecnicamente, ele determina a prioridade do dispositivo no barramento, porém na prática as prioridades exatas não fazem muita diferença. Alguns sistemas selecionam o disco com o número de destino mais baixo como o disco de inicialização e outros exigem que o disco de inicialização seja o de número 0.

Se tiver sorte, um dispositivo terá um botão giratório por meio do qual o número de destino pode ser configurado. Outras maneiras comuns de configurar o número de destino são jumpers e DIP switches. Se não for óbvio como configurar o número de destino num dispositivo, consulte o manual de hardware. Hoje, a maioria das especificações de hardware pode ser encontrada nos sites Web do fabricante; tentar configurar um disco aleatório costumava envolver um processo de tentativa e erro.

O padrão SCSI suporta uma forma de subendereçamento chamada "número de unidade lógica" (LUN, do inglês *logical unit number*). Cada destino possui várias unidades lógicas internas. Um exemplo plausível poderia ser um array de drives com vários discos, mas somente um controlador SCSI. Porém, as unidades lógicas são raramente utilizadas na vida real. Quando você escutar "número da unidade lógica SCSI", pode assumir que é realmente um número de destino que está em discussão até provem o contrário. Se um dispositivo SCSI contiver somente uma unidade lógica, o LUN normalmente terá como padrão 0.

Os barramentos SCSI são em geral relativamente fáceis de serem configurados, porém uma série de coisas pode dar errado:

Snow666

- Muitas estações de trabalho possuem dispositivos SCSI internos. Verifique a listagem dos dispositivos atuais antes de reinicializar para adicionar um novo dispositivo.
- Certifique-se de que um controlador SCSI diferencial possua somente dispositivos diferenciais e terminadores diferenciais conectados a ele e certifique-se de que uma cadeia SCSI de terminação simples não contenha quaisquer dispositivos diferenciais. As técnicas diferenciais e de terminação simples são incompatíveis.
- Depois de ter acrescentado um dispositivo SCSI novo, verifique a listagem dos dispositivos descobertos pelo kernel do Linux quando ele reinicializa para garantir que tudo o que você espera esteja realmente lá. A maioria dos drivers SCSI não detectará dispositivos múltiplos que possuem o mesmo endereço SCSI, que é uma configuração ilegal. Conflitos de endereço SCSI podem levar a um comportamento muito estranho.
- Alguns dispositivos de expansão (que contêm uma fonte de alimentação e um ou mais dispositivos SCSI) terminam o barramento dentro do dispositivo. Se os dispositivos forem conectados após o dispositivo de expansão, você poderá ter problemas de confiabilidade com qualquer um dos dispositivos na cadeia SCSI. Sempre verifique e confira novamente o fato de ter exatamente dois terminadores e que ambos estejam nos finais do barramento.
- O botão giratório usado para configurar o endereço SCSI de um dispositivo é algumas vezes conectado ao contrário. Quando isso acontece, o botão giratório mudará o endereço SCSI, mas não o valor exibido.
- Ao calcular o comprimento de um barramento SCSI-2, certifique-se de contar os cabos internos e os dispositivos de expansão. Eles podem ser relativamente longos. Da mesma forma, lembre-se de que o comprimento máximo pode ser reduzido caso dispositivos SCSI mais antigos sejam adicionados a um barramento SCSI mais novo.
- Jamais se esqueça de que um controlador SCSI usa um dos endereços SCSI!

A interface IDE

O IDE, também chamado de ATA (de AT Attachment), foi desenvolvido para ser simples e barato. Normalmente ele é mais encontrado em PCs ou estações de trabalho de baixo custo. O controlador é embutido no disco, o que reduz os custos de interface e simplifica o firmware. O IDE se tornou popular no final dos anos 80. Logo depois, foi desenvolvido o ATA-2 para satisfazer a demandas crescentes por parte dos consumidores e de fabricantes de unidades de disco.

O ATA-2 adiciona os modos PIO (Programmed I/O, entrada/saída programada) e DMA (Direct Memory Access, acesso direto à memória) e estende os recursos de Plug-and-Play do barramento. Ele também fornece um recurso chamado LBA (Logical Block Addressing, endereçamento de blocos lógicos) que (em combinação com uma BIOS de PC mais sofisticada) suplanta uma limitação que impedia as BIOS de acessar além dos primeiros 1.024 cilindros de um disco. Essa restrição limitava anteriormente os tamanhos dos discos a 504 MB. Quem poderia imaginar que um disco rígido fosse ficar tão grande?

Já que a BIOS gerencia parte do processo de bootstrapping, foi necessário, numa certa época, criar uma pequena partição inicializável dentro dos primeiros 1.024 cilindros para garantir que o kernel pudesse ser carregado por uma BIOS antiga. Assim que o kernel estivesse ativo e em funcionamento, a BIOS não era mais necessária e poderíamos acessar o restante do disco. Essa manobra boba é desnecessária em hardware moderno, pois o LBA se livra do endereçamento CHS (Cylinder-Header-Sector) em favor de um esquema de endereçamento linear.

O ATA-3 acrescenta maior confiabilidade, gerenciamento de energia mais sofisticado e recursos de automonitoramento. O Ultra-ATA é uma tentativa de cobrir o intervalo deixado entre o ATA-3 e o ATA-4, acrescentando modos de alto desempenho chamados Ultra DMA/33 e Ultra DMA/66 que ampliam a largura de banda do barramento de 16 MB/s para, respectivamente, 33 ou 66 MB/s. O ATA-4 também é uma tentativa muito necessária para mesclar o ATA-3 ao ATAPI (ATA Packet Interface), um protocolo que permite que unidades de fita e de CD-ROM funcionem num barramento IDE. Os mais recentes acréscimos à família, ATA-5 e ATA-6, incluem gerenciamento de desempenho otimizado e manipulação de erros que aumentam o desempenho, especialmente num ambiente multiusuário como o Linux.

Os discos IDE são quase sempre usados internamente (a menos que consideremos um disco pendurado fora do computador como "externo" para fins de testes). O comprimento máximo do cabo para um barramento ATA-2 são meros 4,5 m, o que pode dificultar o acesso até mesmo ao compartimento de unidades superior de seu sistema. Além do comprimento curto do cabo, um barramento IDE é capaz de acomodar somente dois dispositivos. Para compensar essas limitações, a maioria dos fabricantes fornece mais de um barramento IDE em suas placas-mãe.

Snow666

Os dispositivos IDE são acessados de uma maneira conectada, o que significa que somente um dispositivo por vez pode estar ativo. Portanto, o desempenho é melhor caso você distribua os dispositivos em vários barramentos. Coloque dispositivos mais rápidos como discos rígidos num barramento e unidades de fita e de CD-ROM num outro para evitar que os dispositivos mais lentos atrapalhem os mais rápidos. Historicamente, o SCSI tem manipulado vários dispositivos num barramento melhor do que o IDE, porém as unidades ATA-5 e ATA-6 estão começando a competir na mesma altura.

O conector IDE é um conector de 40 pinos que conecta a unidade à placa de interface por meio de um flat cable. Padrões IDE mais novos como o Ultra DMA/66 usam um cabo diferente que dispõe de mais pinos terra e, conseqüentemente, reduz o ruído elétrico. Se um cabo ou unidade não estiverem encaixados, certifique-se de que o pino 1 na unidade vá para o pino 1 da placa de interface. O pino 1 normalmente está marcado com um pequeno "1" no lado do conector. Caso ele não esteja marcado, uma regra prática é que o pino 1 normalmente se encontra mais próximo do conector de alimentação. O pino 1 em um flat cable é usualmente identificado pela cor vermelha. Caso não haja essa faixa vermelha em um cabo, basta se certificar de que o cabo esteja orientado para que o pino 1 esteja conectado ao pino 1 e identifique-o claramente para a próxima vez.

Caso tenha mais do que um dispositivo num barramento IDE, você tem de designar um como o mestre e os demais como escravos. Algumas unidades IDE mais antigas não gostam de ser escravas e, portanto, caso esteja tendo dificuldades em fazer uma determinada configuração funcionar, tente inverter os seus papéis e fazer com que o outro dispositivo seja o escravo. Se isso não funcionar, talvez seja melhor tornar cada dispositivo o mestre de seu próprio barramento IDE.

Ao considerar o emprego de hardware IDE, tenha em mente os seguintes pontos:

- Unidades IDE novas funcionam em placas mais antigas e unidades IDE antigas funcionam em placas mais novas. Naturalmente, apenas os recursos comuns a ambos os dispositivos serão suportados.
- O comprimento do cabo é excessivamente curto, o que pode tornar complicado o ato de adicionar um dispositivo extra ao barramento. Caso passe por problemas estranhos, verifique o comprimento do cabo. Um cabo de qualidade pode ser o diferencial.
- Lidar com uma BIOS antiga que não detecta espaço em disco além dos 500 MB é um pesadelo. Verifique se o fabricante lançou uma atualização de firmware para corrigir o problema. Em caso negativo, você poderá substituir a placa-mãe do sistema por US$ 200[3] ou menos; vale a pena. Felizmente, as BIOS tão antigas são uma raridade nos dias de hoje.
- Drivers bem projetados podem aumentar o desempenho e a confiabilidade significativamente, em especial quando eles suportam os padrões mais recentes. Descubra quais recursos o seu driver suporta (por exemplo, modos DMA e PIO) e compare esses recursos com a gama de opções disponíveis no mercado. Esse também é um bom conselho ao comprar discos.

O que é melhor, SCSI ou IDE?

Essa é uma pergunta freqüentemente feita em discussões sobre que vantagens apresentam cada um desses padrões. Porém, você entrará num emaranhado que o levará à seguinte resposta: SCSI é melhor. Normalmente.

Uma resposta mais precisa seria a de que o SCSI bate o IDE em qualquer senso técnico, porém pode ser que o equipamento SCSI apresente um custo que não valha a pena. Para uma estação de trabalho monousuária, um bom disco IDE é uma solução barata, simples e de alta capacidade que fornece 85% do desempenho de uma configuração SCSI. Na maioria dos casos, atualizar uma estação de trabalho monousuária para SCSI não aumentará o desempenho percebido do sistema.

Entretanto, em algumas situações, o SCSI é recomendável ou até mesmo obrigatório:

- Se você precisa absolutamente do melhor desempenho possível, parta para SCSI. Parte do aumento em termos de desempenho provém da superioridade técnica do SCSI, porém uma parte ainda maior virá do fato de os fabricantes de unidades de disco usarem a partilha IDE/SCSI para ajudá-los a estratificarem o mercado de unidades de disco. Algumas unidades IDE podem até ser superiores em termos de desempenho máximo, contudo o SCSI quase sempre fornece um desempenho sustentado melhor.

---

3. A preços do mercado brasileiro, em outubro de 2003, o preço de uma placa-mãe para consumidor final era em torno de R$ 300,00 (N.R.T.).

Snow666

- Os servidores e sistemas multiusuário requerem SCSI. O protocolo SCSI não tem paralelos em termos de habilidade no gerenciamento de várias solicitações simultâneas de uma maneira eficiente. Num sistema congestionado, observaremos uma melhoria concreta e mensurável em termos de desempenho.
- Caso queira conectar vários dispositivos, o SCSI ganha novamente. Os dispositivos SCSI se dão bem com os demais; os dispositivos IDE monopolizam e lutam pelo barramento.
- Talvez seja necessário algum recurso em particular que somente o SCSI forneça. Por exemplo, é um verdadeiro desafio (embora não mais impossível) construir um array de discos que pode ser conectado com o sistema no ar por meio da unidades IDE.

8.2 Geometria do Disco

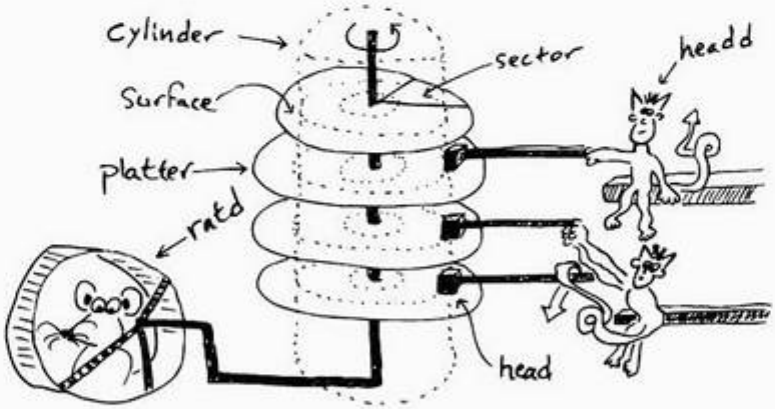
A geometria de um disco rígido e a terminologia usada para se referir às suas várias partes são mostradas no Diagrama B. Estas informações são fornecidas basicamente para melhorar seu conhecimento geral. As unidades de disco modernas ainda se apóiam no mesmo projeto básico, porém o software não mais sabe (ou precisa saber) muito a respeito da construção física da unidade.

Uma unidade de disco típica é formada por lâminas em rotação cobertas com um filme magnético. Os dados são lidos e gravados por uma pequena cabeça que muda a orientação das partículas magnéticas na superfície das lâminas. As lâminas de dados são completamente seladas para que nenhuma sujeira ou pó seja capaz de entrar. Esse recurso torna os discos rígidos fixos muito mais confiáveis do que a mídia removível.

Nos primórdios do hardware de computadores, as unidades de disco normalmente tinham uma lâmina. Um aumento na capacidade de armazenamento foi fornecido por um aumento no diâmetro da lâmina. Na parede de um dos recintos de nossos usuários havia um disco antigo com mais de 10 cm de diâmetro capaz de armazenar aproximadamente 280 KB de dados. Isso é praticamente 25% da capacidade de um disquete moderno.

Hoje, as unidades de disco rígido possuem várias lâminas pequenas empilhadas umas sobre as outras em vez de uma única e grande lâmina. Ambos os lados são usados para armazenar dados, embora um dos lados da lâmina normalmente contenha informações sobre posicionamento e não pode ser usado para armazenamento. As densidades de lâminas simples estão atualmente por volta dos 30 GB, com nenhum final em vista para a Lei de Moore.[4]

Diagrama B Lição sobre geometria de discos



As lâminas giram numa velocidade constante. Elas são lidas e gravadas por pequenas cabeças deslizantes que se movem para a frente e para trás como a agulha num toca-discos. As cabeças flutuam muito próximas da superfície das lâminas, mas não as tocam realmente. A distância entre a cabeça e a lâmina em rotação pode ser comparada a um jato de combate F-16 voando a plena velocidade a 3 m do solo. Se uma cabeça tocar uma lâmina, este evento é chamado de "head crash"; ele pode ser muito destrutivo.

As velocidades rotacionais aumentaram dramaticamente ao longo do tempo. Os discos mais antigos giravam a 3.600 RPM ou 5.40 RPM. 7.200 RPM é atualmente o padrão superior no mercado de armazenamento de massa e 10.000 RPM e 15.000 RPM vêm se tornando populares no mercado de ponta. Velocidades maiores diminuem a latência e aumentam a largura de banda da transferência de dados, mas podem potencialmente introduzir problemas

4. A Lei de Moore afirma que a tecnologia (velocidade de CPU, tamanhos de discos etc.) dobra a cada 18 meses.

Snow666

térmicos originados da maior produção de calor. Certifique-se de ter uma circulação de ar adequada caso planeje adquirir uma unidade de disco de ponta.

É necessário pelo menos uma cabeça para cada superfície. As cabeças em unidades antigas tinham de percorrer grandes distâncias, porém a geometria moderna de lâminas pequenas empilhadas é mais eficiente. O diâmetro dos discos continua a diminuir, de um padrão de 14" (35,5 cm) 20 anos atrás, para 5 ¼" (13,33 cm) 10 anos atrás, para 3 ½" (8,89 cm) ou menos hoje em dia.

Movimentar a cabeça para a posição correta para ler um determinado trecho de dados é chamado de busca. Cada posição que uma cabeça pode ocupar à medida que o disco gira é chamada de trilha. As trilhas são ainda divididas em setores, que têm normalmente um comprimento de 512 bytes.

Um conjunto de trilhas em lâminas diferentes que estão à mesma distância a partir do eixo é chamado de cilindro. Se todas as cabeças se movimentarem juntas, como é típico na maioria das unidades do mercado de armazenamento de massa, os dados armazenados num único cilindro pode ser lido sem qualquer movimento adicional. Embora as cabeças se movam com incrível velocidade, elas ainda se movimentam de maneira muito mais lenta do que a rotação dos discos. Portanto, qualquer acesso de disco que não requeira que as cabeças busquem uma nova posição será mais rápido.

Os sistemas de arquivos Linux padrão tentam explorar esse fato para melhorar a eficiência. Infelizmente, uma prática agora universal conhecida como setorização por zonas, na qual as trilhas no lado mais externo de uma lâmina contêm mais setores do que as trilhas internas, complica essa questão. Para otimizar totalmente os funcionamentos internos da unidade, informações de layout não são compartilhadas com o software. Em vez disso, um esquema de endereçamento CHS (cylinder-head-sector) é formado para se adaptar ao tamanho do disco. Mesmo hoje, muitas otimizações de sistemas de arquivos se baseiam em completa ficção.

Embora a maior parte de quaisquer valores CHS que são multiplicados para atender ao tamanho do disco pode ser usada, o fabricante terá tipicamente alguns valores sugeridos. Algumas configurações — como aquelas com mais de 1.024 cilindros ou mais de 255 cabeças — podem provocar problemas com alguns sistemas operacionais ou boot loaders. Isso normalmente é um problema em PCs rodando em hardware antigo ou em vários sistemas operacionais.

8.3 Uma Visão Geral sobre o Procedimento de Instalação de Discos

O procedimento para adicionar um novo disco envolve as seguintes etapas:

- Conectar o disco ao computador
- Formatar o disco
- Identificar e particionar o disco
- Estabelecer os volumes lógicos
- Criar sistemas de arquivos Linux dentro de partições do disco
- Configurar a montagem automática
- Configurar o "swapping" em partições de swap

As seções a seguir elaboram essas etapas básicas. Começando na página 110, mostramos um exemplo contendo o processo completo do início ao fim de uma unidade de disco.

Conectando o disco

A maneira pela qual o disco está conectado ao sistema depende basicamente da interface que é usada. O restante é tudo questão de montagem e cabeamento. Se o disco for IDE, tente configurar o sistema somente com um disco IDE por barramento. Verifique bem a orientação dos cabos e as configurações mestre/escravo em cada disco. Se o disco for SCSI, verifique bem se você efetuou as terminações apropriadas em ambas as extremidades do barramento SCSI, que o comprimento de cabo seja menor do que o máximo apropriado para a variante SCSI que está usando, e que o novo destino SCSI não conflita com o controlador ou um outro dispositivo no barramento. Para maiores detalhes, veja as seções específicas de sistema mais para o final deste capítulo.

Antes de podermos acessar um disco novo, precisamos de arquivos de dispositivos em **/dev** que apontem para ele. O Linux cria automaticamente arquivos para todos os dispositivos SCSI possíveis. Veja a página 171 para obter maiores informações sobre arquivos de dispositivos.

É possível destruir um sistema de arquivos em segundos gravando aleatoriamente no disco, de modo que devemos configurar permissões bem restritivas para acesso a arquivos de dispositivos de disco. Permitimos acesso de leitura e de gravação para o proprietário (root) e acesso de leitura para o proprietário do grupo (operador); essa

Snow666

configuração permite que **dump** seja executado por operadores sem a necessidade de privilégios de superusuário, porém impede meros mortais de ler do dispositivo bruto.

Formatando o disco

Fornecedores extremamente espertos normalmente divulgam as capacidades de disco em termos do número de bytes não formatados. Cerca de 10% dessa capacidade é usada tipicamente para marcar as superfícies do disco para que o hardware e o software possam localizar os dados que se encontram gravados ali. Ao comprar discos, sempre pense em termos de tamanho formatado e compare os preços de acordo.

Um outro truque comum é indicar os tamanhos dos discos em "megabytes" que são realmente milhões de bytes. No contexto de sistemas computacionais, um megabyte é, na verdade, 2^{20} ou 1.048.576 bytes. Usar a convenção 1 MB = 1.000.000 bytes aumenta a capacidade de uma unidade de disco em cerca de 5%.[5] Certifique-se de verificar suas unidades de medida ao comparar as capacidades de discos.

O processo de formatação grava informações de endereços e marcas de temporização nas lâminas para delinear cada setor. Ele também identifica "blocos ruins", imperfeições na mídia que resultam em áreas que não podem ser lidas ou gravadas de forma confiável. Os discos modernos possuem gerenciamento de blocos ruins embutido, de modo que nem você nem o driver precisam se preocupar com isto.[6]

Todos os discos vêm pré-formatados, e a formatação de fábrica normalmente é mais precisa do que qualquer formatação que você possa vir a fazer em campo. É melhor evitar formatação de baixo nível caso ela não seja realmente necessária. Caso encontre erros de leitura ou de gravação num disco, verifique primeiramente cabos, terminações e problemas de endereçamento, capazes de provocar sintomas similares àqueles de um bloco ruim. Se após esse procedimento você ainda estiver convencido de que o disco está ruim, talvez seja melhor substituí-lo por um novo do que ficar aguardando longas horas para uma formatação terminar.

Os discos IDE normalmente não são projetados para serem formatados fora de fábrica. Entretanto, talvez você seja capaz de obter software de formatação especial do fabricante, normalmente para Windows. Certifique-se de que o software seja adequado para a unidade que pretende formatar e siga cuidadosamente as instruções do fabricante.

Os discos SCSI se formatam em resposta a um comando que enviamos do computador host. O procedimento para envio desse comando varia de sistema para sistema. Em PCs, normalmente podemos dar o comando pela BIOS do controlador SCSI.

Vários utilitários nos permitem verificar a integridade de um disco gravando padrões aleatórios nele e depois os lendo de volta. Esse processo consome muito tempo, a menos que suspeitemos que o disco esteja ruim ou formos cobrar por hora, devemos pulá-lo. Excetuando isso, faça com que os testes sejam efetuados de madrugada. Não se preocupe com "desgastar" um disco com testes agressivos ou uso excessivo. Os discos são projetados para suportar atividade constante.

Identificando e particionando o disco

Depois de um disco ter sido formatado e seus setores ruins remapeados, ele precisa ser dividido em blocos chamados partições. O particionamento permite ao disco ser tratado como um grupo de áreas de dados independentes em vez de uma vasta expansão de blocos. O particionamento também permite que itens de "bonificação" como os blocos de boot e a tabela de partição em si sejam ocultos de software de alto nível (por exemplo, o sistema de arquivos). Somente o driver de dispositivos sabe a respeito do layout do disco inteiro; outro software funciona com a abstração limpa de partições.

As partições tornam mais fáceis os backups, impedem que usuários invadam ilicitamente o espaço de disco de terceiros, melhoram o desempenho e limitam o potencial de programas fora de controle. A tabela de partição é mantida no disco num registro chamado label. O label normalmente ocupa os primeiros blocos do disco. Seu conteúdo exato varia, porém ele geralmente contém informações suficientes para fazer com que o sistema seja inicializado.

As partições são, conceitualmente, distintas e separadas umas das outras. Entretanto, o Linux também define uma partição (a não-numerada, por exemplo, **/dev/sda**) para ser uma imagem de todo o disco. Dessa maneira, os comandos em nível de usuário poderão acessar o disco "diretamente" através de um dispositivo de arquivo comum. Por exemplo, um processo em nível de usuário pode gravar o label do disco ou duplicar seus conteúdos para um disco de backup usando o comando **dd**. Obviamente, essa partição especial tem de ser usada cuidadosamente, já que ela permite que qualquer outra partição no disco seja danificada de uma só vez.

---

5. Obviamente, o prefixo "mega" realmente significa "milhão", de modo que a prática não é inteiramente indefensável.
6. Entretanto, quaisquer blocos ruins que apareçam depois de um disco ter sido formatado não serão "manipulados"; eles podem se manifestar na forma de erros de leitura e de gravação e de dados perdidos.

Snow666

Alguns sistemas vão mais além dessa traiçoeira trajetória e nos permitem definir vários conjuntos de partições sobrepostos. Por exemplo, as partições 1, 2 e 3 podem dividir o disco de uma maneira, enquanto as partições 4 e 5 poderiam fazê-lo de uma outra maneira. Espera-se que se use um conjunto de partições autoconsistentes e simplesmente que se ignore os demais. Na vida real, tais partições sobrepostas são um convite a erros de operadores e são uma causa comum de corrupção aleatória de dados.

Os sistemas modernos tendem a usar menos partições do que seus predecessores, porém na maioria dos sistemas teremos pelo menos as três a seguir:

- **A partição-raiz**: Tudo o que é necessário para acionar o sistema no modo de monousuário é mantido aqui. Uma segunda cópia desta partição é normalmente armazenada num outro disco para fins de emergência.
- **A partição de swap (intercâmbio)**: Uma área de swap armazena páginas de memória virtual quando não há memória física suficiente para armazená-las. Todo sistema deveria ter ao menos uma partição de swap. Consulte a página 587 para obter maiores informações sobre memória virtual.
- **A partição de usuário**: Diretórios de usuários, arquivos de dados, bibliotecas de código-fonte e outros arquivos de dados são colocados aqui.

As opiniões diferem sobre a melhor maneira de dividir discos em partições. Eis algumas dicas:

- Se tiver vários discos, faça uma cópia do sistema de arquivos raiz em um deles e verifique se consegue inicializar por meio dele.
- À medida que for acrescentando memória a uma máquina, você também deve adicionar espaço de swap. Para uso comum, devemos ter pelo menos espaço de swap igual ao da memória física. Essa quantidade de swap permite que um dump de crash de kernel caiba na área de swap no caso de uma pane no sistema.
- Dividir o espaço de swap entre vários discos aumenta o desempenho. Essa técnica funciona para sistemas de arquivos também; coloque os de maior atividade em discos diferentes. Consulte a página 591 para notas sobre este assunto.
- Se pretende fazer backup de alguma partição, não a torne maior do que a capacidade de seu dispositivo de backup. Consulte a página 122.
- Tente agrupar informações que mudam com rapidez em algumas poucas partições que são jogadas em backup freqüentemente.
- É uma boa idéia criar um sistema de arquivos separado (**/tmp**) para arquivos temporários, pois ele limita os arquivos a um tamanho finito e evita que você tenha de fazer backup deles.
- Já que os arquivos de log são mantidos em **/var**, é uma boa idéia que **/var** seja uma partição de disco separada. Deixar **/var** como parte de uma pequena partição-raiz fica mais fácil preencher a raiz e parar a máquina.

Estabelecendo volumes lógicos

O LVM do Linux (Logical Volume Manager, gerenciador de volume lógico) é um subsistema opcional que fornece uma espécie de versão supermodificada do particionamento de disco. Ele permite o agrupamento de discos individuais em "grupos de volumes". A capacidade agregada de um grupo de volume pode então ser alocada a volumes lógicos que são acessados como dispositivos de blocos comuns. O gerenciamento de volumes lógicos nos permite fazer o seguinte:

- Usar e alocar armazenamento em disco de maneira mais eficiente.
- Mover volumes lógicos entre diferentes dispositivos físicos.
- Aumentar e diminuir o tamanho de volumes lógicos com o sistema no ar.
- Capturar "instantâneos" de sistemas de arquivos inteiros.
- Substituir unidades on-line sem interrupção de serviços.

Os componentes de um volume lógico podem ser colocados juntos de diversas maneiras. A concatenação mantém juntos os blocos físicos de cada dispositivo e coloca os dispositivos em ordem um após o outro. Striping intercala os componentes para que blocos virtuais adjacentes sejam, na verdade, espalhados através de vários discos físicos. Reduzindo os gargalos de um único disco, o striping pode normalmente oferecer uma largura de banda maior e uma latência menor.

Snow666

Quando este livro foi escrito, o LVM ainda estava em desenvolvimento ativo, tendo recentemente sido incorporado ao kernel 2.4.x. Talvez o LVM ainda não forneça a solidez de recursos para gerenciamento de volumes com os quais os administradores de sistemas estão acostumados em outras plataformas. [7] Caso esteja interessado em aprender sobre os desenvolvimentos mais recentes do LVM ou planeja instalá-lo, vá ao endereço www.sistina.com/products_lvm.htm.

Embora ainda não seja suportado pelo LVM, o "espelhamento" é mais um outro recurso comum de gerenciadores de volumes. Um volume espelhado é associado a um outro volume do mesmo tamanho. Toda vez que dados forem gravados num dos lados do espelho, ele será duplicado no outro.

As leituras são divididas entre os volumes para aumentar o desempenho. Se um volume passar por uma falha de hardware, o sistema automaticamente passa o controle a um outro volume sem interrupção de serviços. Assim que o problema original tiver sido solucionado, os dois lados do espelho têm de ser sincronizados novamente.

Sistemas de arquivos do Linux

O suporte de sistema de arquivos do Linux evoluiu rapidamente ao longo dos últimos anos à medida que o Linux tem absorvido recursos de uma série de outros sistemas operacionais. O Linux usava originalmente o código de sistema de arquivos do sistema operacional MINIX de Andrew S. Tanenbaum. Porém, o sistema de arquivos MINIX era bastante limitado: ele restringia o tamanho máximo de arquivo a 64 MB e os nomes de arquivos a 14 caracteres. Rapidamente ficou claro que o Linux precisaria passar a suportar vários formatos de sistemas de arquivos e uma camada VFS (Virtual File System) foi introduzida no kernel para suportar este recurso. A camada VFS é particularmente útil em PCs, pois ela fornece a estrutura necessária para montar sistemas de arquivos "nativos" como o infame sistema de arquivos FAT do MS-DOS.

O Second Extended File System, comumente conhecido como ext2fs, é o sistema de arquivos é mais usado com Linux hoje em dia. Ele foi concebido e implementado basicamente por Rémy Card, Theodore Ts'o e Stephen Tweedie. Embora o código para ext2 tenha sido escrito especificamente para Linux, ele adota muitos conceitos e tipos de arquivos do sistema de arquivos FFS do BSD concebido e implementado por Kirk McKusick e equipe em 1984.

O Third Extended File System, ext3fs, é uma extensão admirável do ext2fs originalmente desenvolvido por Stephen Tweedle. Ele acrescenta capacidade de registro de ações (journaling) ao código ext2fs existente, uma modificação conceitualmente simples que fornece funcionalidade extraordinária. As extensões ext3fs foram implementadas transparentemente sem afetar a estrutura fundamental do ext2fs. Na realidade, ainda podemos montar um sistema de arquivos ext3fs como um sistema de arquivos ext2fs — ele simplesmente não terá os recursos de registro de ações habilitados.[8]

O sistema de arquivos ext3fs separa uma área do disco para o arquivo de registro de ações. Quando uma operação de sistema de arquivos ocorre, as modificações necessárias são primeiramente gravadas no arquivo de registro de ações. Quando a atualização do registro de ações é completada, um registro de complemento ("commit record") é gravado para marcar o final da entrada. Somente então o sistema de arquivos normal é modificado. Se ocorresse um travamento no sistema, o log de registros de ações poderia ser usado para reconstruir um sistema de arquivos perfeitamente consistente.

O registro de ações reduz o tempo necessário para execução de verificações de consistência de sistema de arquivos (consulte a seção 8.5) para aproximadamente um segundo por sistema de arquivos. Excetuando-se algum tipo de falha de hardware, o estado de um ext3fs pode ser determinado e restaurado quase que instantaneamente.

ReiserFS é um outro sistema de arquivos para Linux com boas perspectivas de êxito. Como o ext3fs, ele também possui um sistema de arquivos com registro de ações e, portanto, é capaz de manter a consistência do sistema de arquivos apesar de incidentes como quedas de sistema e reinicializações não planejadas (que podem ser comuns num ambiente laptop ou desktop). A versão atual do ReiserFS não é otimizada para sistemas (como servidores) que podem vir a experimentar um grande número de leituras e gravações simultâneas. Uma nova versão, o Reiser4, deve ser lançada em breve.

Além dos recursos de registro de ações, o Reiser4 fornecerá uma interface de sistema de arquivos modular que permitirá aos desenvolvedores de aplicativos e aos administradores de sistemas especificarem como os arquivos devem ser manipulados (e protegidos) de uma forma muito detalhada. Esse recurso apresenta o potencial de aumentar a segurança de arquivos em ambientes especializados.

---

7. Sistemas de armazenamento alternativos, como o LVM, têm de ser avaliados intensamente num ambiente de testes antes de serem usados em servidores de produção. A corrupção de dados pode freqüentemente trazer efeitos colaterais catastróficos.
8. Mas não experimente isso em casa.

www.snow666.blogspot.com
Snow666

8.4 Os Sistemas de Arquivos Ext2 e Ext3

Mesmo depois de um disco rígido ter sido dividido conceitualmente em partições, ele ainda não está pronto para armazenar arquivos. O sistema de arquivos precisa adicionar um pouco do seu custo próprio antes de o disco estar pronto para uso. Nesta seção, descrevemos primeiramente o ext2fs e depois algumas das extensões do ext3fs.

Para instalar um sistema de arquivos dentro de uma partição de disco, use **mke2fs**. A menos que esteja fazendo algo estranho, você deve ser capaz de construir o sistema de arquivos especificando nada mais do que o comando **mke2fs** e o nome da partição.

Veremos o procedimento exato de maneira mais detalhada posteriormente; o restante desta seção discute como os arquivos são colocados no disco ao construirmos o sistema de arquivos. Caso não esteja interessado, simplesmente pule para a próxima seção e se deleite num êxtase de ignorância.

Um sistema de arquivos ext2fs consiste em cinco componentes estruturais:

- Um conjunto de células de armazenamento inode
- Um conjunto de "superblocos" distribuídos
- Um mapa dos blocos de disco no sistema de arquivos
- Um resumo do emprego de blocos
- Um conjunto de blocos de dados

Cada partição de sistema de arquivos é dividida em grupos de blocos. Estruturas como tabelas de inode são alocadas entre os grupos de blocos para que os blocos que são acessados juntos possam ser armazenados próximos uns dos outros no disco. Esse agrupamento reduz a necessidade de ficar procurando por todo o disco ao acessar blocos do mesmo arquivo.

Inodes são entradas de tabelas de comprimento fixo cada uma das quais armazenando informações sobre um arquivo. Já que o espaço para inodes tem de ser separado quando o Linux faz sua estruturação inicial do sistema de arquivos, temos de decidir com antecedência quantos deles criar. É impossível prever exatamente quantos arquivos (inodes) serão necessários algum dia; o Linux usa uma fórmula empírica para conjeturar um número apropriado, baseado no tamanho da partição e num tamanho médio de arquivo.

Podemos ajustar o número de inodes tanto para cima como para baixo ao criarmos o sistema de arquivos: mais inodes para sistemas de arquivos com grande quantidade de arquivos pequenos (como partições de notícias Usenet ou repositórios de código-fonte) e menos inodes para sistema de arquivos com um número menor de arquivos grandes (como, por exemplo, um sistema de arquivos contendo um banco de dados).

Um superbloco é um registro que descreve as características do sistema de arquivos. Ele contém informações sobre o comprimento de um bloco de disco, o tamanho e localização das tabelas de inodes, o mapa de blocos de disco e informações sobre utilização, o tamanho dos grupos de blocos e alguns outros poucos parâmetros importantes do sistema de arquivos. Pelo fato de danos ao superbloco poderem apagar algumas informações extremamente cruciais, várias cópias dele são mantidas em localizações espalhadas (no início de cada grupo de bloco).

Para cada sistema de arquivos montado, o Linux mantém tanto uma cópia em memória do superbloco como várias cópias em disco. A chamada de sistema **sync** despeja os superblocos que estavam armazenados em cache para suas localizações permanentes em disco, tornando o sistema de arquivos consistente num piscar de olhos. Esse salvamento periódico minimiza o nível de dano que ocorreria caso a máquina caísse quando o sistema de arquivos ainda não tivesse atualizado os superblocos. **sync** também descarrega os inodes modificados e blocos de dados em cache. O comando **update** é normalmente executado no momento do boot e dispara o daemon **bdflush**, que executa um **sync** a cada 30 segundos para minimizar a quantidade de perda de dados na eventualidade de uma queda de sistema.

Um mapa de blocos do disco de um sistema de arquivos é uma tabela dos blocos livres que ele contém. Quando são gravados arquivos novos, este mapa é examinado para planejar um esquema de layout eficiente. Os resumos de emprego dos blocos registram informações básicas sobre os blocos que já se encontram em uso.

Configurando extensões ext3fs

Um sistema de arquivos ext3fs é simplesmente um sistema de arquivos ext2fs com um arquivo para registro de ações e suas opções associadas. Na realidade, em sistemas em que ext3fs está disponível, é possível converter um sistema de arquivos ext2fs existente para um sistema de arquivos ext3fs através de **tune2fs**. Se tivéssemos, por exemplo, um sistema de arquivos ext2fs em **/dev/hda4**, poderíamos convertê-lo com

```
# tune2fs -j /dev/hda4
```

Snow666

Precisaríamos então modificar a entrada correspondente em **/etc/fstab** para ler `ext3` em vez de `ext2` (veja o item *Configurando montagem automática*, para maiores informações sobre o arquivo **fstab**). [9]

Observe que sistemas de arquivos ext3 se comportam de forma ligeiramente distinta dos sistemas de arquivos ext2. Uma das modificações mais notáveis é que em vez de sincronizar os blocos de disco não gravados a cada 30 segundos, ext3 faz a sincronização a cada 5 segundos. Também temos a opção de três compromissos diferentes entre desempenho e funcionalidade ao selecionarmos um modo de registro de ações:

- O modo-padrão, "ordered" (ordenado), garante que o sistema de arquivos sempre seja consistente e, como resultado, os arquivos jamais serão corrompidos por uma queda de sistema. Esta é a melhor opção para a maioria dos ambientes.
- O modo "writeback" pode resultar em arquivos corrompidos após uma queda de sistema, porém em alguns casos é mais rápido. Esta é, na verdade, a maneira como funciona a maioria dos sistemas de arquivos com registro de ações e não é substancialmente mais arriscada do que o comportamento-padrão de ext2.
- O modo "journal" (registro de ações) usa um arquivo para registro de ações maior, que pode vir a retardar a recuperação na reinicialização, porém pode ser mais rápido quando usado com um aplicativo de banco de dados.

O modo sistema de arquivos ext3 é configurado com uma entrada `data=`*mode* no arquivo **fstab**, onde *mode* é um dos modos: `ordered`, `writeback` ou `journal`.

Configurando montagem automática

Um sistema de arquivos tem de ser montado antes de se tornar disponível para processos Linux. O ponto de montagem para um sistema de arquivos pode ser qualquer diretório, porém os arquivos e subdiretórios abaixo dele não serão acessíveis enquanto um sistema de arquivos não for montado ali. Veja a seção 5.2, no Capítulo 5, para obter maiores informações sobre montagem de sistemas de arquivos. Certifique-se de que a partição que esteja montando ou verificando possua um sistema de arquivos válido nela — se tentar usar **mount** ou **fsck** em sua partição de swap, você realmente poderá bagunçar as coisas.

Após instalar um disco novo, você deve montar manualmente os novos sistemas de arquivos para testar se tudo está funcionando corretamente. Por exemplo,

```
# mount /dev/sda1 /mnt
```

montaria o sistema de arquivos na partição representada pelo arquivo de dispositivos **/dev/sda1** no diretório **/mnt**. Se o sistema de arquivos for novinho em folha, o seu conteúdo deve ficar parecido com o seguinte:

```
# ls /mnt
lost+found
```

O diretório **lost+found** é criado automaticamente ao criarmos um sistema de arquivos. Ele é usado por **fsck** em emergências; não o remova. O diretório **lost+found** possui algum espaço extra pré-alocado de modo que **fsck** poderá armazenar aí arquivos "desassociados" sem ter de alocar entradas de diretório adicionais num sistema de arquivos instável.

Podemos verificar o tamanho de um sistema de arquivos por meio do comando **df**. Eis um exemplo:

```
% df /home
Filesystem   1k-blocks      Used  Available   Use%  Mounted on
/dev/sda5    17212156   1445776   14892044     9%  /home
```

As unidades relatadas por **df** são blocos de 1K ou podemos usar a opção **--block-size** para especificarmos nossas unidades.

Um arquivo **fstab** que inclua o sistema de arquivos acima poderia ser parecido com o seguinte:

```
/dev/sdb6    /           ext2    defaults        1 1
/dev/sda1    /boot       ext2    defaults        1 1
/dev/sda5    /home       ext2    defaults        1 2
/dev/cdrom   /mnt/cdrom  iso9660 noauto,owner,ro 0 0
/dev/fd0     /mnt/floppy auto    noauto,owner    0 0
```

9. Caso esteja convertendo seu sistema de arquivos raiz, também precisará informar seu boot loader (LILO ou GRUB) para carregar os módulos ext3. Veja a página de manual referente a **mkinitrd** para maiores detalhes.

Snow666


```
LABEL=/usr   /usr      ext2   defaults      1 2
LABEL=/var   /var      ext2   defaults      1 2
none         /proc     proc   defaults      0 0
/dev/sdb7    swap      swap   defaults      0 0
```

Há seis campos por linha, separados por espaços em branco. Cada linha descreve um único sistema de arquivos. Os campos são tradicionalmente alinhados para fins de legibilidade, porém o alinhamento não é obrigatório.

O primeiro campo dá o nome do dispositivo ou o rótulo que foi associado a ele por **e2label** (o `LABEL=`form). O arquivo **fstab** pode incluir montagens de sistemas remotos, em cujo caso o primeiro campo contém um caminho NFS. A notação *server:/export* indica o diretório */export* na máquina chamada *server*.

Veja o Capítulo 17 para obter maiores informações sobre o NFS.

O segundo campo especifica o ponto de montagem e o terceiro campo indica o tipo de sistema de arquivos. O nome do tipo exato usado para identificar sistemas de arquivos locais varia, dependendo da configuração do sistema. Tradicionalmente, o Linux com as configurações-padrão tem usado ext2, porém nos dias atuais o Red Hat e o SuSE fornecem a opção de selecionarmos ext3 como sistema de arquivos local padrão.

No caso do sistema de arquivos **/proc** acima, o `none` na primeira coluna é um valor substituto. O tipo de sistema de arquivos virtual é especificado pelo valor `proc` na terceira coluna; esse sistema de arquivos exporta várias informações do kernel para utilitários de usuários.

O quarto campo lista as opções de montagem (`rw`, de read/write (leitura/gravação), é o padrão). O quinto campo especifica um valor de "freqüência de descarga" que pode ser usado teoricamente por produtos de backup, mas normalmente não é usado.

***fsck** é descrito na seção 8.5.*

O sexto campo especifica a passagem na qual **fsck** deve verificar o sistema de arquivos. Sistemas de arquivos com o mesmo valor nesse campo são verificados concorrentemente, se possível. Não configure dois sistemas de arquivos no mesmo disco com o mesmo valor; caso contrário você fará com que a cabeça de leitura do disco fique indo e vindo de tal modo que o desempenho será degradado significativamente. Apenas sistemas de arquivos em discos separados devem ser verificados em paralelo.

mount, **umount**, **swapon** e **fsck** todos lêem o arquivo **fstab**, de maneira que é importante que os dados apresentados aqui sejam corretos e completos. **mount** e **umount** usam o arquivo **fstab** para descobrir o que você quer que seja feito caso especifique somente um nome de partição ou ponto de montagem na linha de comandos. Por exemplo, usando o arquivo **fstab** mostrado, o comando

```
# mount /mnt/cdrom
```

seria o mesmo que digitar

```
# mount -t iso9660 -o ro,noauto,owner /dev/cdrom /mnt/cdrom
```

O comando **mount -a** monta todos os sistemas de arquivos regulares listados no arquivo **fstab**; ele é usualmente executado por meio dos scripts de inicialização no momento do boot. O flag **-t** restringe a operação a sistemas de arquivos de um certo tipo. Por exemplo,

```
# mount -at ext2
```

montaria todos os sistemas de arquivos de disco ext2 num sistema Linux. O comando **mount** lê **fstab** seqüencialmente; como conseqüência, os sistemas de arquivos montados sob outros sistemas de arquivos devem seguir as partições de seus pais no arquivo **fstab**. Por exemplo, a linha para **/var/log** deve seguir a linha para **/var** se **/var** for um sistema de arquivos distinto.

O comando **umount** para desmontar sistemas de arquivos aceita uma sintaxe similar. Na maioria dos sistemas, não podemos desmontar um sistema de arquivos que um processo está usando como seu diretório atual ou um no qual os arquivos estão abertos. Há maneiras de contornar essa restrição; veja o Capítulo 5.

Habilitando o swapping

Uma das primeiras vantagens do UNIX era sua implementação de memória virtual. Esse recurso permite ao sistema operacional fingir que tem mais memória do que realmente possui. Se processos tentarem usar essa memória "extra", os discos do sistema serão usados como um tipo de RAM ultralenta. Jogar o conteúdo da memória de e para o disco é conhecido como swapping ou paginação. [10]

10. Swapping e paginação são tecnicamente distintos. Por enquanto, os agrupamos e chamamos a combinação de "swapping". Para uma descrição mais detalhada de memória virtual no Linux, veja o Capítulo 26.

www.snow666.blogspot.com
Snow666

Veja o Capítulo 26 para obter maiores informações sobre divisão de áreas de swap.

Para tornarmos o swapping eficiente, normalmente usaríamos partições brutas (partições sem sistemas de arquivos) como armazenamento-reserva. Em vez de usar uma estrutura de sistema de arquivos para controlar o conteúdo da área de swap, o kernel mantém seu próprio mapeamento simplificado de blocos de memória para blocos de disco. Também é possível fazer swap para um arquivo numa partição com sistema de arquivos, porém esta configuração é mais lenta do que usar uma partição dedicada.

Quanto mais espaço de swap tivermos, mais memória virtual nossos processos poderão alocar. O melhor desempenho de swapping é atingido quando a área de swap é dividida entre várias unidades (ou melhor ainda, entre vários barramentos SCSI).

Podemos habilitar o swapping manualmente para um determinado dispositivo, porém em geral você vai querer ter esta função executada automaticamente no momento do boot. Na maioria dos sistemas, áreas de swap podem ser listadas no arquivo **fstab**, o mesmo arquivo que é usado para enumerar sistema de arquivos montáveis. Uma entrada de swap se parece com o seguinte:

```
/dev/sdb7        swap        swap        defaults        0 0
```

Durante a inicialização, o comando **swapon** é executado para habilitar o swapping em todas as partições de swap listadas no arquivo **fstab**.

8.5 FSCK: Verificação e Reparo de Sistemas de Arquivos

O sistema de arquivos tradicional do Linux (ext2fs) é surpreendentemente confiável e ele realiza um trabalho extraordinário no enfrentamento de quedas de sistema inesperadas e hardware com comportamento estranho. Porém, os sistemas de arquivos podem se tornar danificados ou inconsistentes de uma série de maneiras.

Qualquer hora que o kernel entra em pânico ou a alimentação falha, podem ser introduzidas pequenas inconsistências nos sistemas de arquivos que estavam ativos imediatamente antes da pane. Já que o kernel armazena em buffers tanto blocos de dados como informações de resumo, a imagem mais recente do sistema de arquivos é dividida entre o disco e a memória. Durante uma pane, a parte da imagem em memória é perdida. Os blocos colocados em buffer são efetivamente "sobrescritos" com as versões que foram salvas em disco mais recentemente.

Há uma série de abordagens para corrigir esse problema. Danos menores podem normalmente ser resolvidos com o comando **fsck** ("filesystem consistency check", ou seja, verificação da consistência do sistema de arquivos). Essa não é uma forma arquitetonicamente muito elegante de abordar o problema, porém ela funciona muito bem para todas as inconsistências comuns.

Sistemas de arquivos com recursos de registro de ações como o reiserFS e ext3fs gravam metadados num arquivo de log seqüencial que é despejado para o disco antes do retorno de cada comando. Os metadados eventualmente migram do log para sua residência permanente dentro do sistema de arquivos. Se o sistema trava, o log pode ser retornado até o ponto de consistência mais recente; não é necessária uma checagem cruzada completa do sistema de arquivos. **fsck** ainda é executado no momento do boot para garantir que o sistema de arquivos se encontra num estado consistente, porém ele roda muito mais rápido quando comparado com uma verificação de sistema de arquivos ext2. Esse recurso pode nos poupar horas de tempo de boot num sistema com grandes sistemas de arquivos.

Se alguma forma de registro de ações não estiver disponível, temos de aguardar o **fsck** fazer os seus procedimentos. Os cinco tipos mais comuns de danos são os seguintes:

- Inodes não referenciados
- Contagens de links inexplicavelmente grandes
- Blocos de dados não utilizados não registrados nos mapas de blocos
- Blocos de dados listados como livres e que também são usados num arquivo
- Informações de resumo incorretas no superbloco

O **fsck** pode corrigir esses cinco problemas de maneira segura e automática. Se o **fsck** fizer correções num sistema de arquivos, devemos executá-lo novamente até que o sistema de arquivos se torne completamente ordenado.

Os discos são normalmente verificados no momento do boot através de **fsck -p**, que examina todos os sistemas de arquivos locais listados em **/etc/fstab** e corrige os cinco erros listados acima. O Linux mantém controle de quais sistemas de arquivos foram desmontados de modo ordenado e verifica somente os "sujos". Se alguma forma de registro de ações estiver habilitada, o **fsck** simplesmente nos informa que o sistema de arquivos está registrado e retorna o log ao último estado consistente.

www.snow666.blogspot.com
Snow666

fsck -p também pode ser executado em um determinado sistema de arquivos. Por exemplo:

```
# fsck -p /dev/sda5
```

Quando **fsck -p** lê o arquivo **fstab** para descobrir quais sistemas de arquivos devem ser verificados, ele obedece a seqüência indicada pelo último campo de cada entrada. Os sistemas de arquivos são verificados em ordem numérica crescente. Se dois sistemas de arquivos estiverem em discos diferentes, eles podem receber o mesmo número de seqüência; essa configuração faz com que **fsck** verifique-os simultaneamente, minimizando o tempo gasto na espera de I/O de disco. Sempre verifique primeiro a partição-raiz.

Erros que não caem nas cinco categorias anteriores são potencialmente graves. Eles fazem com que **fsck -p** solicite ajuda e depois saia. Nesse caso, execute **fsck** sem a opção **-p**. Quando executado em modo manual, **fsck** lhe solicita que confirme cada um dos reparos que ele pretende fazer. A lista a seguir mostra alguns dos erros que **fsck** considera perigosos o bastante para garantirem a intervenção do operador.

- Blocos reivindicados por mais de um arquivo
- Blocos reivindicados fora do intervalo do sistema de arquivos
- Contagens de links muito pequenos
- Blocos não contabilizados
- Diretórios que se referem a inodes não alocados
- Vários erros de formato

Infelizmente, é difícil corrigir um disco manualmente sem ter conhecimento profundo da implementação do sistema de arquivos. A menos que esteja tentando se estabelecer como um líder religioso numa cidadezinha do Texas, você jamais deve gravar diretamente no sistema de arquivos através de arquivos de dispositivos.

Na prática, esse estado de coisas significa que temos poucas opções exceto a de aceitar as correções sugeridas por **fsck**. Podemos minimizar os problemas registrando cuidadosamente as mensagens produzidas por **fsck**, já que algumas vezes elas fornecem uma dica sobre o arquivo ou arquivos que estão provocando problemas. Se **fsck** solicitar permissão para eliminar um arquivo, você deve tentar copiá-lo para um sistema de arquivos diferente antes de permitir que **fsck** prossiga. Esteja atento ao fato de que cada vez que você tenta acessar um sistema de arquivos danificado, corre o risco de provocar uma pane no sistema.

Consulte o Capítulo 10 para obter maiores informações sobre dump e outras maneiras de fazer backup de sistemas de arquivos.

Se um sistema de arquivos danificado (um que o **fsck** não é capaz de reparar automaticamente) contém dados valiosos, não fique experimentando antes de fazer um backup consistente. Você pode tentar fazer um **dump** para o disco, mas já que **dump** tem como expectativa ler um sistema de arquivos não danificado, a imagem resultante pode ter falta de dados (ou então **dump** pode travar). A melhor política de segurança é aplicar **dd** ao disco inteiro para um arquivo de backup ou disco de backup.

Se **fsck** localiza um arquivo cujo diretório-pai não pode ser determinado, ele coloca o arquivo no diretório **lost+found** no nível mais alto do sistema de arquivos. Já que o nome dado a um arquivo é registrado somente no diretório-pai do arquivo, os nomes para arquivos órfãos não estarão disponíveis e os arquivos substituídos em **lost+found** serão nomeados com seus números de inode.

8.6 Adicionando um Disco ao Linux: um Guia Passo a Passo

Depois de seu novo disco ser instalado, é uma boa idéia certificar-se de que o sistema consiga detectar o novo dispositivo antes de fazer o boot no kernel. Se ele for um disco IDE, faça uma verificação para garantir que o disco seja reconhecido na tela de configuração da BIOS, que normalmente podemos acessar digitando uma seqüência de teclas mágicas antes de o sistema inicializar. Consulte os manuais que vêm com o seu computador ou placa-mãe para obter informações específicas sobre configuração de BIOS para dispositivos IDE. Na maioria dos casos, não é necessária nenhuma configuração especial.

Muitas placas SCSI também possuem uma tela de configuração de BIOS que podemos acionar antes de o sistema ser inicializado. Se essa opção estiver disponível, você poderá varrer o barramento SCSI para ter certeza de que o novo dispositivo apareça. Se esse procedimento parar ou produzir uma mensagem de alerta, é possível que você tenha escolhido um ID de SCSI que já estivesse em uso ou que você não instalou terminadores nos lugares corretos.

Você também pode usar a BIOS do SCSI para aplicar uma formatação de baixo nível num disco. Essa operação leva um longo tempo em alguns discos e não pode ser interrompida; portanto, planeje com antecedência.

Snow666

Veja, no Capítulo 12, maiores informações sobre como instalar drivers de dispositivo.

Se uma placa SCSI não fornece uma interface com o usuário própria, sempre podemos tentar inicializar o sistema e verificar as mensagens exibidas pelo kernel. Caso não veja nenhuma mensagem de um driver SCSI, é possível que você precise instalar o driver antes de o disco poder ser reconhecido pelo kernel.

Em nosso caso, observamos as seguintes mensagens de nosso adaptador host SCSI BusLogic.

```
scsi0 : BusLogic BT-948
scsi : 1 host.
   Vendor: SEAGATE Model: ST446452W     Rev: 0001
   Type: Direct-Access            ANSI SCSI revision: 02
Detected scsi disk sda at scsi0, channel 0, id 3, lun 0
scsi0: Target 3: Queue Depth 28, Asynchronous
SCSI device sda: hdwr sector=512 bytes. Sectors=91923356 [44884 MB] [44.9 GB]
sda: unknown partition table
```

Ignore alertas acerca da tabela de partições, já que esta é a primeira vez que o disco foi usado. Assim que o sistema tiver terminado a inicialização, você poderá prosseguir para o processo de particionamento do disco.

Devemos primeiramente verificar se os arquivos de dispositivos para o disco já existem (eles devem existir). No Linux, os nomes para os arquivos de dispositivos de disco SCSI assumem a forma **/dev/sd**XN, onde X é uma letra minúscula que identifica o drive (**a** é o disco SCSI de menor numeração, **b** é o segundo menor e assim por diante[11]) e N é o número de partição iniciando em 1. Ao referir-se ao disco todo, simplesmente omita o número da partição. Não há nenhum dispositivo de disco (bruto) do tipo caractere (character device) no Linux.

Neste exemplo, nosso disco é o primeiro na cadeia SCSI. A primeira partição é portanto **/dev/sda1** e o disco como um todo é referido como **/dev/sda**. Se esses arquivos de dispositivos não existissem, poderíamos criá-los com o script **/dev/MAKEDEV**:[12]

```
# cd /dev
# ./MAKEDEV sda
```

O disco agora está pronto para ser particionado. Como acontece na maioria dos sistemas operacionais para PCs, a ferramenta usada para partição no Linux é chamada de **fdisk**. Embora todas as versões de **fdisk** façam praticamente a mesma coisa (elas implementam o sistema de partição padrão da Microsoft), há muitas variações entre elas. É aconselhável você ler o manual referente ao seu sistema em particular para garantir que ele atenda àquilo que mostramos aqui.

```
# fdisk /dev/sda
The number of cylinders for this disk is set to 5721.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
```

Já que usaremos somente este disco em nosso sistema Linux, ignoraremos o alerta acima. Conforme afirmado na discussão da página 99, algumas vezes é importante tornar a primeira partição pequena para garantir que ela funcionará com uma BIOS antiga e que funcionará com outros sistemas operacionais que poderiam vir a ser instalados no sistema.

O programa **fdisk** é interativo; pressionando-se **m** é exibida uma lista de todos os seus comandos. Os que usamos aqui são os seguintes:

- **n** ou **new** para criar uma partição nova
- **t** ou **type** para alterar o tipo de uma partição
- **p** ou **print** para imprimir a tabela de partições
- **w** ou **write** para gravar a tabela de partições no disco

11. Note que essa letra refere-se à *ordem* dos números de destino dos dispositivos SCSI, não aos números em si. Caso adicione ou retire um disco, todas as letras mudarão!
12. **MAKEDEV** é padrão em sistemas Red Hat e Debian, porém você precisará usar **mknod** em sistema SuSE. Veja a página 171 para maiores informações sobre a criação de arquivos de dispositivos.

Snow666

Já que nosso disco não tem ainda partições, iniciaremos com a criação de uma. Se houver partições antigas de discos anteriores, você terá de removê-las com o comando **delete** de **fdisk**, antes de criar a sua. O programa **fdisk** não modificará nada até que o informemos para gravar a tabela de partições.

A tabela de partições possui espaço para quatro partições "primárias" que podem ser usadas para armazenar dados. Alternativamente, podemos criar uma partição "estendida", que é uma partição primária que aponta para uma outra tabela de partições, nos dando outras quatro partições "lógicas". Embora o emprego de partições estendidas possa suplantar a restrição normal de quatro partições, é mais simples adotar as partições primárias caso somente algumas forem necessárias e é isto que faremos neste caso:

```
Command (m for help): new
Command action
   e  extended
   p  primary partition (1-4): p
Partition number (1-4): 1
First cylinder (1-5721, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-5721, default 5721): +2G

Command (m for help): print
Disk /dev/sda: 255 heads, 63 sectors, 5721 cylinders
Units = cylinders of 16065 * 512 bytes

   Device  Boot  Start  End   Blocks  Id  System
/dev/sda1            1  255  2048256  83  Linux
```

Criamos uma partição de swap de forma similar, exceto pelo fato de mudarmos o tipo de Linux para Linux swap. Embora o kernel não se preocupe com o tipo de partição, alguns programas ou scripts usam o tipo para descobrir o que é cada partição. As áreas de swap não podem atualmente exceder a 2 GB, de modo que especificamos o tamanho máximo. Isso provavelmente seja um exagero para a maioria dos aplicativos, porém já que temos espaço em disco para gastar, podemos muito bem ser generosos. **mkswap** nos alertará caso ele não possa usar todo o espaço que alocamos.

```
Command (m for help): new
   e  extended
   p  primary partition (1-4): p
Partition number (1-4): 2
First cylinder (256-5721, default 256): 256
Last cylinder or +size or +sizeM or +sizeK (256-1275, default 1275): 511
Command (m for help): type
Partition number (1-4): 2
Hex code (type L to list codes): 82
Changed system type of partition 2 to 82 (Linux swap)
```

A terceira partição, que contém o restante do disco, é definida de maneira similar. Vejamos a tabela de partições mais uma vez antes de gravá-la.

```
Command (m for help): print
Disk /dev/sda: 255 heads, 63 sectors, 5721 cylinders
Units = cylinders of 16065 * 512 bytes
   Device  Boot  Start   End    Blocks  Id  System
/dev/sda1            1   255   2048256  83  Linux
/dev/sda2          256   511   2056320  82  Swap
/dev/sda3          512  5721  41849325  83  Linux
```

Um asterisco aparece próximo ao número de blocos caso a partição não termine num limite de cilindro. Podemos eliminar a partição e recriá-la introduzindo um número de cilindros (conforme ilustrado acima) ou então conviver com o fato de que uma pequena quantidade de espaço em disco não possa ser utilizada. Estamos satisfeitos com a nova tabela de partições, de modo que podemos gravá-la no disco.

```
Command (m for help): write
The partition table has been altered!
Calling ioctl() to re-read partition table.
SCSI device sda: hdwr sector=512 bytes. Sectors=91923356 [44884 MB] [44.9 GB]
sda: sda1 sda2 sda3
Syncing disks.
```

Snow666

Alguns administradores gostam de reinicializar antes de gravar a tabela de partições para garantir que tudo esteja estável e consistente antes de criar sistemas de arquivos. Provavelmente isso não seja necessário nos dias atuais, porém se você já está acostumado a instalar o Windows, talvez se sinta mais à vontade reinicializando.

Agora estamos prontos para criar sistemas de arquivos. Para criar um sistema de arquivos ext2fs tradicional, execute **mke2fs** com o dispositivo e o tamanho da partição na linha de comandos. (O tamanho pode ser encontrado na saída gerada pelo comando **fdisk** anterior e é especificado em blocos.)

```
# mke2fs /dev/sda1 2048256
Linux ext2 filesystem format
514000 inodes, 2048001 blocks
102412 blocks (5.00%) reserved for the super user
First data block=1
Block size=1024 (log=0)
Fragment size=1024 (log=0)
250 block groups
8192 blocks per group, 8192 fragments per group
2056 inodes per group
Superblock backups stored on blocks:
8193, 16385, 24577, 32769, 40961, 49153, 57345, 65537,
...
Writing inode tables: 250/250 done
Writing superblocks and filesystem accounting information:
done
```

O processo para criar o sistema de arquivos maior é o mesmo, mas isso provavelmente leverá mais tempo. Caso saiba que não precisará de todos os inodes que **mke2fs** aloca por padrão, poderá reduzir o número de inodes por grupo, acelerando **mke2fs** e reservando mais espaço para dados efetivos. Da mesma maneira, caso queira aumentar o número de inodes para sistemas de arquivos que abrigarão um grande número de arquivos muito pequenos, é muito melhor ter muitos inodes do que poucos, já que a falta de inodes o impedirá de criar qualquer outro arquivo. Não é possível adicionar mais inodes depois de o sistema de arquivos ter sido criado. Caso você se depare com essa situação, precisará fazer um **dump** dos dados do sistema de arquivos para fita ou para um arquivo numa outra partição, executar novamente **mke2fs** com um número maior de inodes (**-i**) e depois restaurar (**restore**) os dados para a partição. Dias de divertimento!

Caso esteja criando um sistema de arquivos ext3fs, precisará usar **mke2fs** com o flag **-j**, o que criará um sistema de arquivos com recurso de registro de ações adequado para a maioria dos aplicativos. Você também poderá usar **-J** para especificar explicitamente o tamanho do arquivo de registro de ações que residirá no novo sistema de arquivos (**-J size=***x*) ou a identidade de um dispositivo externo que conterá o arquivo de registro de ações (**-J device=***y*). Instalações típicas localizam o arquivo de registro de ações (que deve ter entre 1.024 e 102.400 blocos do sistema de arquivos; no caso da partição recém-criada, com blocos de 1 KB, este tamanho deve ser entre 1 MB e 100 MB) dentro do sistema de arquivos em si.

Executamos **fsck** em nossos sistemas de arquivos para garantir que eles foram criados apropriadamente. O flag **-f** força **fsck** para verificar novos sistemas de arquivos em vez de supor que eles estejam organizados.

```
# fsck -f /dev/sda1
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sda1: 11/514000 files (0.0% non-contiguous), 67014/2048001 blocks
```

Novos sistemas de arquivos podem ser montados assim que seus pontos de montagem forem criados:

```
# mkdir /bkroot
# mount /dev/sda1 /bkroot
# df /bkroot
Filesystem    1k-blocks    Used    Available    Use%    Mounted on
/dev/sda1       1981000      13      1878575      0%    /bkroot
```

Para garantirmos que o sistema de arquivos monte os novos sistemas de arquivos no momento da inicialização, adicionamos uma linha para cada um deles ao arquivo **/etc/fstab**. Cada linha deve listar o nome do dispositivo, o

Snow666

ponto de montagem, o tipo de sistema de arquivos, as opções de montagem, a freqüência de backup e o número de passagem para **fsck**:

```
/dev/sda1      /bkroot      ext2      defaults      0 1
/dev/sda3      /new         ext2      defaults      0 2
```

Um boot loader tem de ser gravado no dispositivo de disco para torná-lo inicializável. Dependendo da instalação, o comando **lilo** ou **grub** faz a efetiva instalação. Veja o Capítulo 2 para maiores informações sobre como configurar e instalar um boot loader.

A etapa final é criar o espaço de swap e adicioná-lo ao sistema. As partições de swap têm de ser inicializadas com **mkswap**, que usa como argumentos o nome do dispositivo e o tamanho da partição de swap em setores[13]. O tamanho pode ser obtido por meio de **fdisk**, como mostrado anteriormente. Também podemos passar o flag **-c** para **mkswap**, que o informa para verificar o dispositivo à procura de blocos defeituosos (*bad blocks*) antes de usá-los. Porém, isto leva um bom tempo, e realmente não é necessário.

Com a área de swap criada, nós a habilitamos com o comando **swapon. swapon** também verifica que a área de swap tenha sido adicionada apropriadamente.

```
# mkswap -c /dev/sda2 2056320
Setting up swapspace version 1, size = 2105667584 bytes
# swapon /dev/sda2
# swapon -s
Filename   Type      Size     Used   Priority
/dev/hda5  partition 133020   688    -1
/dev/sda2  partition 2056316  0      -2
```

Como acontece com sistemas de arquivos regulares, temos de adicionar a nova partição de swap ao arquivo **/etc/fstab** caso queiramos que o sistema se lembre dela na próxima reinicialização. A entrada a seguir seria apropriada para nosso disco de exemplo:

```
/dev/sda2    none[14]    swap     defaults     0 0
```

Certifique-se de reinicializar para testar as mudanças que foram feitas para o arquivo **/etc/fstab** e para certificar-se de que os novos sistemas de arquivos e espaço de swap entrem on-line de forma correta.

8.7 EXERCÍCIOS

E8.1 Liste todas as variações de SCSI. Quais conectores vão com quais variantes? Ignorando-se as diferenças nos conectores, quais são os problemas de compatibilidade entre as diversas versões SCSI?

E8.2 Qual a diferença entre formatar e particionar um disco? Qual a diferença entre particionar e criar um sistema de arquivos?

★ **E8.3** Usando recursos impressos ou da Internet, identifique os drives IDE e SCSI de melhor desempenho. Os benchmarks usados para avaliar esses drives refletem a maneira pela qual um servidor Linux ocupado usaria o seu disco de inicialização? Quanto em termos de custo adicional você pagaria pelo SCSI e quanto de melhoria em termos de desempenho (se efetivamente alguma) você obteria pelo investimento adicional?

★ **E8.4** Adicione um disco ao seu sistema. Crie uma partição no disco novo como uma partição-raiz de backup; instale um kernel e faça a inicialização nele. Mantenha um arquivo como registro de todas as etapas necessárias para completar esta tarefa. Talvez você ache o comando **script** útil. (Requer acesso root.)

★ **E8.5** O que é um superbloco e para que ele é usado? Pesquise a definição da estrutura de superblocos ext2fs nos arquivos de cabeçalho de kernel e discuta o que representa cada um destes campos na estrutura.

★ **E8.6** Liste todos os comandos e argumentos que você usaria para criar um sistema de arquivos num disco em cada uma das seguintes circunstâncias.

a) O disco será usado como armazenamento para diretórios de usuários.
b) O disco será usado como espaço de swap.
c) O disco armazenará a fila de e-mails numa grande fonte de spams.
d) O disco armazenará um banco de dados Oracle.

---

13. O tamanho da partição de swap não precisa ser informado para o comando **mkswap**; ele é mantido, e aceito pelo comando, apenas para fins de compatibilidade (N.R.T.).
14. Conforme a página de manual do arquivo **fstab**, a entrada para uma partição de swap deve conter "none" como ponto de montagem da partição, já que ela não é montada, mas sim habilitada (N.R.T.).

Snow666

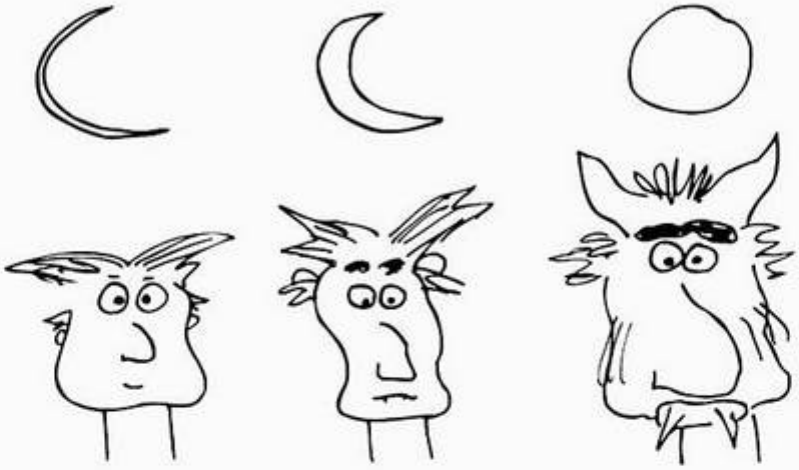
★★ **E8.7** Que campos são armazenados num inode num sistema de arquivos ext2fs? Liste o conteúdo do inode que representa o arquivo **/etc/motd**. Onde o nome de arquivos do arquivo é armazenado? (Ferramentas como **hexdump** e **ls -i** talvez ajudem.)

★★ **E8.8** Examine o conteúdo de um arquivo de diretórios com um programa como **hexdump** ou **od**. Cada registro de comprimento variável representa um arquivo nesse diretório. Examine a estrutura de entradas de diretórios e explique a função de cada campo usando um exemplo de um arquivo de diretório real. A seguir, examine o diretório **lost+found** em qualquer sistema de arquivos. Por que há tantos nomes lá quando o diretório **lost+found** parece estar vazio?

★★★★★ **E8.9** Escreva um programa que percorra o sistema de arquivos e imprima o conteúdo dos arquivos **/etc/motd** e **/etc/termcap**. Mas não abra os arquivos diretamente; abra o arquivo dispositivo bruto para a partição-raiz e use as chamadas de sistema **seek** e **read** para decodificar o sistema de arquivos e localizar os blocos de dados apropriados. **/etc/motd** normalmente é pequeno e provavelmente contém somente blocos diretos. **/etc/termcap** provavelmente requeira que você decodifique blocos indiretos. Dica: ao ler os arquivos de cabeçalho de sistema, certifique-se de ter localizado a estrutura de inodes em disco do sistema de arquivos, e não a estrutura de inodes em memória. (Requer acesso root.)

Snow666

9 Processos Periódicos



O segredo para manter-se no controle de um sistema é automatizar o maior número possível de tarefas. Por exemplo, um programa **adduser** é capaz de adicionar usuários de forma mais rápida que nós e com uma chance menor de cometer erros. Praticamente qualquer tarefa pode ser codificada num script **expect**, Perl ou shell.

Normalmente é mais útil ter um script ou comando executado sem qualquer intervenção humana. Poderíamos, por exemplo, querer que um script verificasse (digamos, de meia em meia hora) se os roteadores e bridges de uma rede estão funcionando corretamente e fazer com que esse script envie um e-mail para nós quando forem descobertos problemas[1].

9.1 CRON: Agendando Comandos

Sob o Linux, a execução periódica é normalmente tratada pelo daemon **cron**. **cron** é iniciado quando da inicialização do sistema e permanece em execução enquanto o sistema estiver no ar. **cron** lê um ou mais arquivos de configuração contendo listas de linhas de comandos e horários nos quais eles são invocados. As linhas de comandos são executadas por **sh**, de maneira que qualquer coisa que podemos fazer manualmente também pode ser feita por meio de **cron**.[2]

cron surgiu originalmente na árvore da família UNIX nos anos 70, e versões tradicionais contêm uma série de recursos estranhos e não tão úteis na prática. O Linux é distribuído com uma versão conhecida como "Vixie-cron", em homenagem ao seu autor, Paul Vixie. Vixie-cron é uma releitura moderna que fornece maior funcionalidade sem toda a confusão anterior.

Um arquivo de configuração **cron** é chamado "crontab", abreviatura de "cron table". **cron** procura arquivos crontab em três locais: **/var/spool/cron** (ou num subdiretório de **/var/spool/cron** chamado tabs no SuSE e crontabs no Debian), **/etc/cron.d** e **/etc/crontab**.

Arquivos crontab por usuário são armazenados no diretório **/var/spool/cron**. Tipicamente, há (pelo menos) um arquivo crontab por usuário: um para root, um para jsmith e assim por diante. Os arquivos crontab são nomeados com os nomes de login dos usuários a quem pertencem, e **cron** usa esses nomes de arquivos para descobrir qual UID

---

1. Muitas instalações vão além disso e discam para um pager com um modem para chamar um administrador de sistema assim que for detectado algum problema.
2. Na verdade, **/bin/sn** sob o Linux é realmente um link para o shell **bash**, uma versão otimizada (e reimplementada) do shell Bourne encontrado em sistemas UNIX. É possível configurar **cron** para usar outros shells também.

Snow666

usar ao executar os comandos que cada arquivo contém. O comando **crontab** transfere arquivos crontab de e para esse diretório.

Arquivos crontab que agendam tarefas de manutenção de sistema e outras tarefas definidas pelo administrador de sistemas são armazenados no arquivo **/etc/crontab** e em outros arquivos localizados no diretório **/etc/cron.d**. Esses arquivos possuem um formato ligeiramente diferente dos arquivos crontab de usuário, pois eles permitem que comandos sejam executados como um usuário arbitrário. **cron** trata as entradas **/etc/crontab** e **/etc/cron.d** exatamente da mesma maneira. Em geral, **/etc/crontab** é proposto como um arquivo para o administrador de sistemas manter manualmente, ao passo que **/etc/cron.d** é fornecido como um local onde pacotes de software podem instalar quaisquer entradas crontab que eles possam vir a precisar.

Quando **cron** é iniciado, ele lê todos os seus arquivos de configuração, os armazena na memória e depois entra em estado de dormência. A cada minuto, **cron** desperta, verifica os horários de modificação nos arquivos crontab, recarrega quaisquer arquivos que foram modificados e depois executa quaisquer tarefas programadas para aquele instante antes de retornar ao estado de dormência.

cron não compensa comandos que são perdidos enquanto o sistema está fora do ar ou por alterações abruptas no horário do sistema (por exemplo, caso você perceba que o relógio do sistema está 20 minutos atrasado e o reinicialize com o horário correto). Se você usar **cron** para tarefas sensíveis a horários, talvez queira investigar **anacron**, uma versão de horário contínuo que é fornecida em sistemas Red Hat e Debian. Ele é conceitualmente muito similar a **cron**, porém ele garante que comandos não sejam pulados devido à aparente descontinuidade no fluxo do tempo.

Por razões que não são claras, **cron** foi renomeado para **crond** em sistemas Red Hat.

9.2 O Formato de Arquivos Crontab

Todos os arquivos crontab num sistema compartilham um formato similar. Comentários são introduzidos com um sinal # na primeira coluna de uma linha. Cada linha não comentada contém seis ou sete campos e representa um comando:

```
minuto hora dia mês dia-da-semana [username] comando
```

Os primeiros seis campos são separados por espaço em branco, porém dentro do campo *comando*, espaço em branco é assumido literalmente. *username* é encontrado somente em **/etc/crontab** e nos arquivos do diretório **/etc/cron.d**; ele especifica o usuário como quem o *comando* deve ser executado.

Esse campo não está presente ou é necessário nos arquivos crontab específicos (aqueles armazenados em **/var/spool/cron**) pois o UID é implícito pelo nome de arquivo.

Os campos *minuto*, *dia*, *mês* e *dia-da-semana* informam quando executar o comando. Suas interpretações são mostradas na Tabela 9.1.

Tabela 9.1 Especificações de tempo crontab.

Campo	Descrição	Intervalo
minuto	Minutos da hora	0 a 59
hora	Hora do dia	0 a 23
dia	Dia do mês	1 a 31
mês	Mês do ano	1 a 12
dia-da-semana	Dia da semana	0 a 6 (0 = Domingo)

Cada um dos campos relacionados a tempo podem conter:

- Um asterisco, que coincide com tudo.
- Um inteiro único, que coincide exatamente.
- Dois inteiros separados por um traço, coincidindo com um intervalo de valores.
- Uma série de inteiros ou intervalos separados por vírgulas, coincidindo com qualquer valor listado.

Por exemplo, a especificação de tempo

```
45 10 * * 1-5
```

Snow666

significa "10:45 da manhã, de 2ª à 6ª". Dica: jamais coloque um asterisco no primeiro campo a menos que queira que o comando seja executado a cada minuto.

Há uma ambigüidade potencial a observar com os campos *dia-da-semana* e *dia*. Cada dia é tanto um dia da semana quanto um dia do mês. Se *dia-da-semana* e *dia* são especificados, um dia precisa satisfazer somente uma de duas condições para ser selecionado.

Por exemplo,

```
0,30 * 13 * 5
```

significa "a cada meia hora na 6ª feira e a cada meia hora no 13º dia do mês" e não "a cada meia hora da sexta-feira 13".

Vixie-cron também permite valores incrementais nas especificações de horário de crontab. Por exemplo, as séries 0,3,6,9,12,15,18,21 podem ser escritas de maneira mais concisa em Vixie-cron como 0-21/3.

comando é a linha de comando **sh** a ser executada. Ela pode ser qualquer comando shell válido e não deve ser colocada entre aspas. *comando* é considerado para continuar até o final da linha e pode conter espaços ou tabulações.

*Vá ao Capitulo 3 para obter maiores informações sobre o comando **su**.*

Um sinal de porcentagem (%) é usado para indicar linhas novas dentro do campo *comando*. Somente o texto até o primeiro sinal de porcentagem é incluso no comando em si; as linhas remanescentes são dadas ao comando como entrada-padrão.

Eis alguns exemplos de comandos crontab:

```
echo The time is now `date` > /dev/console
write garth % Hi Garth. % Remember to get a job.
cd /etc; /bin/mail -s "Arquivo de senhas" evi < passwd
```

E eis alguns exemplos completos de entradas crontab:

```
30 2 * * 1 (cd /users/joe/project; make)
```

Essa entrada será ativada às 2:30 a cada segunda-feira pela manhã. Ela executará **make** no diretório **/users/joe/project**. Uma entrada como essa poderia ser usada para iniciar uma longa compilação num horário em que os demais usuários não estariam usando o sistema. Usualmente, qualquer saída produzida por um comando **cron** é enviada para o "proprietário" do crontab. [3]

```
20 1 * * * #find /tmp -atime +3 -exec rm -f { } ';'
```

Esse comando será executado às 1:20 de cada manhã. Ele elimina todos os arquivos do diretório **/tmp** que não foram acessados nos últimos 3 dias.

```
55 23 * * 0-3,6 /staff/trent/bin/acct-script
```

Essa linha executa **acct-script** às 23:55 toda noite exceto às quintas e sextas.

Também é possível especificar variáveis de ambientes e seus valores num arquivo crontab Vixie-cron. Veja a página de manual **crontab**(5) para obter maiores detalhes.

9.3 Gerenciamento de Crontab

crontab *filename* instala *filename* (nome do arquivo) como seu crontab, substituindo qualquer versão anterior. **crontab -e** verifica uma cópia de seu crontab, ativa o seu editor nele (conforme especificado pela variável de ambiente EDITOR) e depois o submete novamente ao diretório crontab. **crontab -l** lista o conteúdo de seu crontab para a saída-padrão e **crontab -r** o elimina, deixando-o sem nenhum arquivo crontab por completo.

A maioria dos sistemas dá espaço para fornecer um argumento *username* para que os crontabs de outros usuários possam ser vistos ou editados. Por exemplo, **crontab -u jsmith –r** apaga o crontab pertencente ao usuário jsmith.

Sem argumentos de linha de comando, **crontab** tentará ler um crontab de sua entrada-padrão. Caso entre acidentalmente nesse modo, não tente sair com <Control-D>; fazer isso apagará o crontab inteiro. Use em seu lugar <Control-C>.

Como padrão, todos os usuários podem submeter arquivos crontab a **cron**. Dois arquivos de configuração, **/etc/cron.deny** e **/etc/cron.allow** (em sistemas SuSE, estes arquivos são chamados **/var/spool/cron/deny** e **/var/spool/cron/allow**), permitem que passemos por cima desse padrão. Se o arquivo allow existir, então ele contém uma lista de todos os usuários que podem submeter crontabs, um por linha. Nenhuma pessoa não listada pode

3. Ou seja, o usuário com base no qual o arquivo crontab recebe o nome. O verdadeiro proprietário de arquivos crontab geralmente é o root.

Snow666

chamar o comando **crontab**. Se o arquivo allow não existe, então o arquivo deny é verificado. Ele, também, é apenas uma lista de usuários, mas o significado é inverso: todo o mundo, exceto os usuários listados, tem acesso permitido. Se não existir nem o arquivo allow nem o arquivo deny, somente root pode submeter crontabs. Em sistemas Debian, se não existirem os arquivos cron.allow e cron.deny, todos os usuários podem submeter crontabs.

É importante notar que o controle de acesso é implementado por **crontab**, não por **cron**. Se um usuário colocar um arquivo crontab no diretório apropriado por outros meios, **cron** executará cegamente os comandos que ele contém.

9.4 Empregos Comuns para Cron

Uma série de tarefas-padrão são especialmente adequadas para serem chamadas por **cron**, e estas normalmente formam o grosso do material no crontab do root. Nesta seção analisamos uma série de tarefas desse tipo e as linhas crontab usadas para implementá-las.

Os sistemas Linux normalmente vêm com entradas crontab pré-instaladas, a maioria em **/etc/cron.d**. Caso queira desativar as entradas-padrão, transforme-as em comentários inserindo um sinal # no início de cada linha. Não as elimine; talvez você precise delas no futuro.

Além do mecanismo **/etc/cron.d**, as distribuições Linux também pré-instalam entradas crontab que executam os scripts num conjunto de diretórios bem conhecidos, fornecendo portanto uma outra forma para os pacotes de software instalarem tarefas periódicas sem a necessidade de editar nenhum arquivo crontab. Por exemplo, scripts em **/etc/cron.daily** são executados uma vez por dia e scripts em **/etc/cron.weekly** são executados uma vez por semana.

Limpando o sistema de arquivos

Alguns dos arquivos em qualquer sistema Linux são puro lixo (os arquivos de sistema não). Por exemplo, toda vez que um programa trava, o kernel cria um arquivo chamado **core** contendo uma imagem do espaço de endereços do programa.[4] Arquivos core são úteis para os desenvolvedores de software, porém para os administradores eles normalmente são uma perda de espaço. Os usuários normalmente nem sabem a respeito dos arquivos core, de modo que eles tendem a não eliminá-los por conta própria.

O NFS, Network File System, é descrito no Capítulo 17.

O NFS é uma outra fonte de arquivos extras. Pelo fato de os servidores NFS serem sem estado, eles precisam usar uma convenção especial para preservar arquivos que foram eliminados localmente, mas que continuam em uso por uma máquina remota. A maioria das implementações renomeia tais arquivos para **.nfs***xxx*, onde *xxx* é um número. Várias situações podem resultar no esquecimento e no abandono desses arquivos após eles supostamente terem sido eliminados.

Muitos programas criam arquivos temporários em **/tmp** ou **/var/tmp** que não são apagados por um motivo ou por outro. Alguns programas, especialmente os editores, adoram criar cópias de backup de cada arquivo com o qual eles trabalham.

Uma solução parcial para o problema de arquivos-lixo é instituir algum tipo de reivindicação noturna de espaço em disco por parte de **cron**. Os sistemas modernos normalmente já vêm com algum tipo desse configurado, porém é uma boa idéia revisar o comportamento-padrão de seu sistema para certificar-se de que é apropriado para o seu caso. Abaixo estão vários exemplos de "limpeza" implementados com o comando **find**.

```
find / -xdev -name core -atime +7 -exec rm -f { } ';'
```

Esse comando elimina imagens de memória que não foram acessadas durante a última semana. O argumento **-xdev** garante que **find** não irá percorrer sistemas de arquivos além do raiz; essa restrição é importante em redes em que muitos sistemas de arquivos podem ser montados de forma cruzada. Caso queira limpar mais do que um sistema de arquivos, use um comando separado para cada um (note que **/var** é tipicamente um sistema de arquivos separado).

```
find / -xdev -atime +3 '(' -name '#*' -o -name '.#*' -o -name '*.CKP' -o
-name '*~' -o -name '.nfs*' ')' -exec rm -f { } ';'
```

Esse comando elimina arquivos iniciados por **#** ou **.#** ou **.nfs** ou que finalizem com **~** ou **.CKP** e que não foram acessados nos últimos três dias. Esses padrões são típicos de várias espécies de arquivos temporários e de backup de editores.

```
cd /tmp; find . ! -name . ! -name lost+found -type d -mtime +3
    -exec /bin/rm -rf { } ';'
```

4. A palavra "core" significa "memória". Esse termo é originário dos primeiros sistemas computacionais que usavam como elementos de memória pequenos anéis de ferrita montados numa malha trançada.

Snow666

Esse comando elimina recursivamente todos os subdiretórios de **/tmp** não modificados nas últimas 72 horas. Arquivos de texto puro em **/tmp** são eliminados no momento do boot do sistema pelos scripts de inicialização de sistema, porém alguns sistemas não eliminam diretórios. Se um diretório chamado **lost+found** existe, ele é tratado especialmente e não é eliminado. Isso é importante caso **/tmp** seja um sistema de arquivos separado.

Se você usar qualquer um desses comandos, deve ter certeza de que os usuários estejam alertas sobre as políticas de limpeza.

Veja o Capítulo 18 para obter maiores informações sobre compartilhamento de arquivos de configuração.

Distribuição via rede de arquivos de configuração

Se estiver rodando uma rede de máquinas, normalmente é conveniente manter uma única versão para toda a rede dos arquivos de configuração como o banco de dados de nomes alternativos de e-mail (normalmente **/etc/mail/aliases**). Versões mestre desses arquivos podem ser distribuídas toda noite com **rsync, rdist** ou um script **expect**.

Algumas vezes, é necessário pós-processamento. Talvez você precise, por exemplo, executar o comando **newaliases** para converter um arquivo de texto de nomes alternativos de e-mail para o formato hashed usado por **sendmail** caso a opção `AuotRebuildAliases` não esteja configurada em seu arquivo **sendmail.cf**. Talvez você também precise carregar arquivos num banco de dados administrativo como NIS.

Rodízio de arquivos log

O Linux realiza um excelente trabalho de gerenciamento de seus arquivos log, porém alguns deles crescem sem limite até serem reinicializados manualmente. Há várias maneiras de impedir que logs estourem; truncá-los em intervalos periódicos é a mais simples delas.

Uma estratégia mais conservadora é a de realizar um "rodízio" dos arquivos log mantendo várias versões mais antigas de cada um deles. Esse esquema evita que os arquivos log fiquem fora de controle, mas jamais o deixa sem informações de log recentes. Já que o rodízio de logs é um evento recorrente e programado regularmente, trata-se de uma tarefa ideal para **cron**. Veja *Rodízio de arquivos log* no Capítulo 11 para obter maiores detalhes.

9.5 Exercícios

E9.1 Um usuário local em seu sistema tem abusado de seus privilégios crontab executando tarefas custosas em intervalos regulares. Depois de ter solicitado a ele várias vezes para parar com esse tipo de procedimento, você se vê forçado a revogar os privilégios deste usuário. Liste as etapas necessárias para eliminar o crontab deste usuário e certificar-se de que ele não adicionará um novo.

★ **E9.2** Imagine três tarefas (além daquelas já mencionadas ao longo deste capítulo) que precisariam ser executadas periodicamente. Crie entradas crontab para cada tarefa e especifique em quais arquivos crontab as entradas devem entrar.

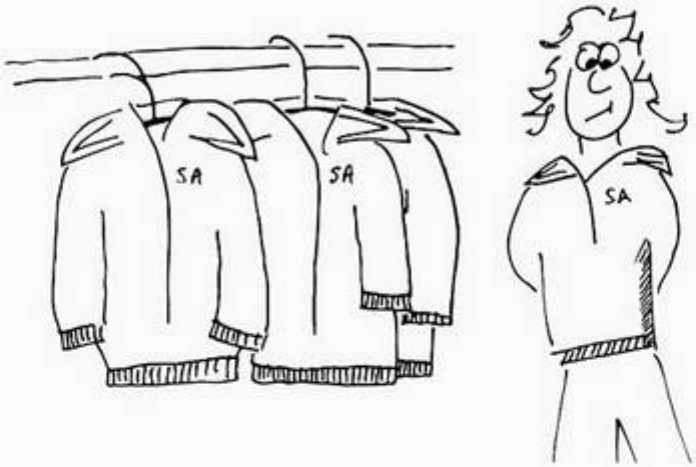
★ **E9.3** Crie um script que mantenha seus arquivos de inicialização (**~/.[a-z]***) sincronizados entre todas as máquinas nas quais você tem uma conta. Programe esse script para ser executado regularmente no **cron**.

★ **E9.4** Localize os arquivos crontab de seu sistema Linux e selecione três entradas. Decodifique cada uma delas e descreva quando é executada, o que faz e por que você imagina que a entrada é necessária. (Requer acesso root.)

★ **E9.5** Usando as páginas de manual para os comandos **du**, **sort** e **head** como referências, crie um script que determine quais dez diretórios de usuários são os maiores no sistema. Programe o script para ser executado toda segunda à meia-noite e faça com que ele remeta esta saída para você. Dica: você vai querer usar uma ordem numérica invertida. (Requer acesso root.)

Snow666

10 Backups



Na maioria das instalações as informações armazenadas nos computadores valem mais do que os próprios computadores. Elas também são muito mais difíceis de serem substituídas. Proteger essas informações é uma das tarefas mais importantes do administrador (e, infelizmente, mais entediantes).

Há centenas de maneiras criativas e não tão criativas de perder dados. Bugs de software rotineiramente corrompem arquivos de dados. Usuários apagam acidentalmente o trabalho de suas vidas. Hackers e funcionários descontentes apagam discos. Problemas de hardware e desastres naturais derrubam salas de máquinas inteiras.

Se executados corretamente, os backups (também conhecidos como "dumps") permitem a um administrador restaurar um sistema de arquivos (ou qualquer parte de um sistema de arquivos) à condição em que ele se encontrava no momento do último backup. Os backups têm de ser realizados cuidadosamente e num cronograma estrito. O sistema de backup e a mídia de backup também precisam ser testados regularmente para verificar se estão funcionando corretamente.

A integridade de fitas de backup afeta diretamente o objetivo de uma empresa. A Gerência Sênior precisa entender o que os backups supostamente devem fazer, e não o que eles *querem* que os backups façam. É aceitável perder um dia de trabalho no Departamento de Computação de uma universidade, porém provavelmente não é aceitável numa empresa de comercialização de commodities.

Começamos este capítulo com alguma filosofia geral sobre backups, seguida de uma discussão dos dispositivos de backup e mídia mais comumente usados (seus pontos fortes, fraquezas e custos). A seguir, discutimos os comandos de backup e arquivamento e damos algumas sugestões sobre quais comandos são mais indicados para que tipos de situação. A seguir falamos sobre como desenvolver um esquema de backups e revisamos a mecânica dos populares utilitários **dump** e **restore**. Finalmente, damos uma olhada no Amanda, um pacote de backup de rede gratuito e depois apresentamos alguns comentários sobre alternativas.

10.1 Práticas Consagradas

Antes de partirmos para os fundamentos dos backups, queremos passar algumas dicas genéricas que foram aprendidas ao longo do tempo (normalmente, a maneira mais difícil). Nenhuma destas sugestões é uma regra absoluta, mas você perceberá que quanto mais delas usar, mais estável será o seu processo de dump.

www.snow666.blogspot.com
Snow666

Execute todos os dumps em uma máquina

Muitos utilitários de backup permitem que realizemos dumps pela rede. Embora haja alguma perda em termos de desempenho ao fazermos isso, o aumento na facilidade da administração vale a pena. Descobrimos que o melhor método é rodar um script em uma localidade central que execute **rdump** (por meio de **rsh** ou **ssh**) em cada máquina onde o backup precise ser feito ou para usar um pacote de software (pago, gratuito ou shareware) que automatiza este processo. Todos os dumps devem ir para o mesmo dispositivo de backup (sem rebobinar, obviamente).

Se sua rede for muito grande para ter o backup feito por uma única unidade de fita, você ainda deve tentar manter o seu sistema de backup o mais centralizado possível. A centralização facilita a administração e nos permite verificar se todas as máquinas tiveram seus backups realizados de maneira correta. Dependendo da mídia de backup que estiver usando, você poderá colocar mais de uma unidade de fita num servidor sem afetar o desempenho. Porém, com as unidades de fita de alto desempenho (12 MB/S ou mais) de hoje, pode ser impraticável fazer isso.

Rotule suas fitas

É essencial que você rotule cada fita de backup de maneira clara e completa. Uma fita sem identificação é uma fita em branco.

As fitas em si devem ser rotuladas para identificar de forma exclusiva seus conteúdos. Informações detalhadas como listas de sistemas de arquivos e datas dos backups podem ser escritas nas tampas.

Você tem de ser capaz de restaurar os sistemas de arquivos raiz e **/usr** sem examinar nos scripts de backup. Identifique as fitas de backup para esses sistemas de arquivos com seus formatos, a sintaxe exata do comando usado para criá-los e quaisquer informações que você precisaria para restaurar por meio delas sem referir-se à documentação on-line.

Há uma abundância de programas para rotulação gratuitos ou não. Poupe uma grande dor de cabeça para você mesmo, investindo em um. Se você compra etiquetas para sua impressora laser, o fornecedor de etiquetas normalmente pode lhe fornecer software (Windows) que gere rótulos.

Selecione um intervalo de backup razoável

Quanto maior for a freqüência com que os backups forem feitos, menor a quantidade de dados que pode ser perdida em um acidente. Porém, os backups usam recursos de sistema e tempo de operador. O administrador de sistemas deve fornecer integridade de dados adequada a um custo razoável de tempo e de material.

Em sistemas atarefados, geralmente é apropriado fazer backups de sistemas de arquivos com diretórios de usuários todo dia de trabalho. Em sistemas que são usados menos intensamente ou no qual os dados são menos voláteis, talvez você decida que realizar backups várias vezes por semana é suficiente. Num sistema pequeno com somente um usuário, realizar backups somente uma vez por semana talvez seja adequado. Qual a quantidade de dados que os seus usuários estão propensos a perder?

Selecione sistemas de arquivos cuidadosamente

Os sistemas de arquivos que raramente são modificados não precisam ter o backup feito com tanta freqüência como os diretórios dos usuários. Se somente alguns poucos arquivos mudarem num sistema de arquivos de outra maneira estático (como **/etc/passwd** no sistema de arquivos raiz), estes arquivos podem ser copiados todo dia para uma outra partição cujo backup é feito regularmente.

Se **/tmp** é um sistema de arquivos separado, ele não deve ter o seu backup feito. O diretório **/tmp** não deve conter nada essencial, de forma que não há nenhuma razão para preservá-lo. No caso de isso parecer óbvio, conhecemos uma grande instalação que realiza backups diários de **/tmp**.

*Consulte o Capítulo 9 para obter maiores informações sobre **cron**.*

Faça backups diários caberem numa fita

Num mundo perfeito, poderíamos fazer backups diários de todos os sistemas de arquivos de usuário numa única fita. Mídia de alta densidade como DLT e AIT tornam esse objetivo prático para algumas instalações. Você pode montar uma fita todo dia antes de deixar o trabalho e realizar os backups durante a madrugada por meio de **cron**. Dessa maneira, os backups ocorrem num horário em que os arquivos provavelmente não ficarão mudando e os backups têm impacto mínimo sobre os usuários.

Infelizmente, esse objetivo está se tornando cada vez menos realístico. Quando os usuários podem comprar discos de 40 GB por US$ 150, não há muito em termos de barreira econômica para impedir o aumento do espaço em disco. Por que limpar seus discos e forçar quotas quando simplesmente podemos gastar um pouco de dinheiro para solucionar o problema?

www.snow666.blogspot.com
Snow666

Se não é possível colocar todos os seus backups diários em uma fita, você tem várias opções:

- Comprar uma unidade de fita de maior capacidade.
- Comprar um empilhador ou biblioteca e alimentar várias mídias num único dispositivo.
- Mudar a seqüência de backup.
- Criar um script mais inteligente.
- Usar vários dispositivos de backup.

Um sistema automatizado de backup deve sempre registrar o nome de cada sistema de arquivos do qual foi feito o backup. O registro cuidadoso sempre nos permite pular diretamente para o sistema de arquivos correto quando queremos restaurar uma fita. Também é uma boa idéia registrar a ordem dos sistemas de arquivos no lado de fora da fita. (Dissemos isto anteriormente, porém vale a pena repetir: certifique-se de usar o dispositivo de fita não rebobinável para gravar fitas com vários backups.)

Torne os sistemas de arquivos menores do que seu dispositivo de backup

dump e outros utilitários prontos para serem usados são perfeitamente capazes de transferir sistemas de arquivos para várias fitas. Porém se um backup ocupar várias fitas, é necessário a presença de um operador para troca das fitas[1] e as fitas têm de ser rotuladas cuidadosamente para possibilitar que as restaurações sejam executadas facilmente. A menos que você tenha uma boa razão para criar um sistema de arquivos realmente grande, não o faça.

Mantenha fitas fora da instalação

A maioria das organizações mantém backups fora do prédio onde o sistema está instalado para que na ocorrência de algum incidente como um incêndio, por exemplo, não sejam destruídos tanto os arquivos originais quanto os backups. "Fora da instalação" pode ser qualquer coisa desde um cofre num banco até a casa do presidente da empresa. As empresas que se especializam em armazenamento seguro de mídias de backup garantem um ambiente seguro e climatizado para os seus arquivos. Certifique-se sempre de que esse tipo de fornecedor de serviços tenha boa reputação, tenha hipoteca e seguro.

A velocidade com que as fitas são transferidas para fora da instalação depende com que freqüência você precisa restaurar arquivos e quanta latência pode aceitar. Algumas instalações evitam tomar essa decisão realizando dois backups diários (em unidades de fita diferentes): uma que fique no local e a outra que seja transferida imediatamente.

Proteja os seus backups

Dan Geer, um consultor em segurança, disse: "O que faz um backup? Ele viola de maneira confiável permissões de arquivos a distância". Hmmm.

Assegure suas fitas de backup. Elas contêm todos os dados de sua empresa e podem ser lidas por qualquer um que tenha acesso físico a elas. Você não somente deve manter suas fitas fora da instalação como também mantê-las a sete chaves. Se você usa uma instalação de armazenamento comercial para essa finalidade, a empresa com quem negocia deve garantir a confidencialidade das fitas sob seus cuidados.

Algumas empresas sentem tão fortemente em relação à importância dos backups a ponto de fazerem duplicatas, o que realmente não é uma má idéia em si.

Limite a atividade durante os backups

A atividade do sistema de arquivos deve ser limitada durante backups, pois mudanças podem fazer com que o utilitário de backup cometa erros. Podemos limitar a atividade fazendo os backups quando poucos usuários estiverem usando o sistema (à noite ou em fins de semana) ou tornando todos os sistemas de arquivos acessíveis somente para o utilitário de backup. (Essa precaução parece boa na teoria, porém raramente é usada na prática. Os usuários querem acesso 24 horas por dia, 7 dias por semana a todos os sistemas de arquivos. Nos dias de hoje é impossível fazer um backup sem nenhuma atividade no disco.)

Verifique a seção 17.5 para obter maiores informações sobre servidores de arquivos.

Os servidores de arquivos como aqueles fabricados pela Network Appliance fornecem backups on-line com instantâneos do sistema de arquivos em intervalos regulares e ajustáveis. Esse recurso permite backups seguros de um sistema de arquivos ativo e é uma das vantagens importantes de usar um servidor de arquivos dedicado.

1. Ou seja, a menos que você tenha um empilhador, jukebox ou biblioteca, e que o seu utilitário de backup o suporte.

www.snow666.blogspot.com
Snow666

Verifique suas fitas

Já ouvimos muitas histórias de terror sobre administradores de sistemas que não viam problemas em seus regimes de backup a não ser depois de uma falha grave no sistema. É essencial monitorar continuamente o seu procedimento de backup e verificar se ele está funcionando corretamente. Erros de operadores arruínam mais backups do que qualquer outra coisa.

A primeira verificação é fazer com que o seu software de backup tente reler fitas imediatamente depois de ele ter finalizado o backup.[2] Varrer uma fita para verificar se ela contém o número de arquivos esperado é uma boa verificação. Seria melhor se todas as fitas fossem varridas, porém isso não parece prático para uma grande empresa que usa centenas de fitas por dia. Uma amostra aleatória seria mais prudente neste ambiente.

*Consulte a seção 10.4 para obter maiores informações sobre **restore**.*

Normalmente é útil gerar uma tabela de conteúdo de cada sistema de arquivos (usuários do **dump** podem usar **restore -t**) e armazenar os resultados em disco. Esses catálogos devem ser nomeados de uma maneira que os associe à fita apropriada; por exemplo, **okra:usr.Jan13**. Uma coletânea de uma semana desses registros torna mais fácil descobrir em que fita um determinado arquivo perdido se encontra. Simplesmente usamos **grep** para o nome do arquivo e selecionamos a instância mais nova.

Além de fornecer um catálogo de fitas, ser bem-sucedido na leitura da tabela de conteúdo da fita é uma indicação razoável de que o backup está OK e que seremos capazes de ler a fita quando precisarmos fazê-lo. Uma rápida tentativa de restaurar um arquivo aleatório nos dará mais confiança em nossa capacidade de restaurar dados daquela fita.[3]

Temos de tentar restaurar periodicamente de várias fitas para termos certeza de que ainda é possível fazê-lo. De vez em quando, tente restaurar de uma fita de backup antiga (meses ou anos). As unidades são conhecidas por saírem de alinhamento ao longo do tempo e de se tornarem incapazes de ler suas fitas antigas. As fitas podem ser recuperadas por uma empresa especializada neste tipo de serviço, porém será caro.

Uma checagem relacionada é verificar se somos capazes de ler as fitas num hardware diferente do nosso. Se a sua sala de máquinas pegar fogo, não adianta muito saber se as fitas poderiam ter sido lidas numa unidade de fita que agora está destruída.

Considere o seguinte caso, que circulou boca a boca alguns anos atrás: *Uma importante empresa de pesquisas da Califórnia tinha um operador que se ocupava tanto fazendo hacking que não tinha tempo para fazer a sua verdadeira função: os backups. Ele abria fitas, as identificava e as arquivava sem jamais ter colocado nenhum dado nela. Essa charada prosseguiu por dois ou três meses até que alguém insistiu em ter um arquivo recuperado. O que aconteceu com ele? Foi despedido? Não, ele foi transferido internamente, porém foi eventualmente preso e acusado por fraudes eletrônicas não relacionadas. Circulavam rumores de que ele pegou 40 anos de cadeia.*

Defina o ciclo de vida de uma fita

As fitas possuem uma vida finita. É fantástico reciclar sua mídia, porém certifique-se de seguir as recomendações do fabricante referentes à vida útil das fitas. A maioria dos fabricantes quantifica essa vida em termos do número de passagens que uma fita pode resistir: um backup, uma restauração e um **mt fsf**, (file skip forward, pula para o primeiro bloco do próximo arquivo), cada uma dessas operações representa uma passagem.

Projete seus dados para backups

Com os discos tão baratos e novas arquiteturas de armazenamento tão confiáveis, é tentador não fazer backup de todos os nossos dados. Uma arquitetura de armazenamento sensata — projetada em vez de criada desordenadamente à medida que surge a necessidade de aumento de espaço em disco — pode colaborar muito para tornar os backups mais tratáveis. Comece fazendo um inventário de suas necessidades de armazenamento:

- Os diversos tipos de dados com os quais sua instalação lida.
- A volatilidade esperada de cada tipo de dado.
- A freqüência necessária para atender a um nível de perda de dados esperado.
- As fronteiras políticas sobre as quais os dados serão disseminados.

Use essas informações para projetar a arquitetura de armazenamento de sua instalação com backups e tendo crescimento potencial em mente. A maioria das instalações não está pronta para depositar total confiança em

2. Podemos usar **restore -C** para verificar uma fita de backup em relação a uma árvore de diretórios.
3. Por exemplo, **restore -t** lê somente a tabela de conteúdo do backup, que é armazenada no início da fita. Quando efetivamente formos restaurar um arquivo, testaremos uma região mais extensa da mídia.

Snow666

"caixas-pretas" chamativas como sistemas RAID ou instantâneos Network Appliance. Preservar diretórios de projetos e diretórios de usuários num servidor de arquivos dedicado pode facilitar o gerenciamento de dados e garantir sua segurança.

Prepare-se para o pior

Depois de ter estabelecido um procedimento de backup, explore a situação do pior caso: sua instalação foi completamente destruída. Determine o nível de dados que serão perdidos e quanto tempo levaria para ter o sistema no ar novamente (incluindo o tempo que levaria para adquirir hardware novo). A seguir, determine se você pode conviver com suas respostas.

10.2 Dispositivos e Mídia para Backup

Já que muitos tipos de falhas podem danificar vários componentes de hardware de uma só vez, os backups devem ser gravados em algum tipo de mídia removível. Por exemplo, fazer o backup do disco rígido para um outro (embora melhor do que nenhum backup em si) fornece um nível de proteção baixo em relação a uma falha de uma controladora. As empresas que fazem seus backups via Internet entraram em cena ao longo dos últimos anos, porém a maioria dos backups ainda é armazenada localmente.

Muitos tipos de mídia armazenam dados usando partículas magnéticas. Essas mídias estão sujeitas a danos pela ação de campos elétricos e magnéticos. Eis alguns danos específicos a serem evitados:

- Alto-falantes possuem grandes eletromagnetos; não é uma boa idéia armazenar fitas sobre ou próximas a eles. Até mesmo alto-falantes pequenos desenvolvidos para uso em computadores podem ser prejudiciais.
- Transformadores e fontes de alimentação (inclusive no-breaks) geram campos eletromagnéticos. Os gabinetes de fontes usadas para alimentar vários periféricos contêm transformadores.
- Os discos rígidos e unidades de fita possuem motores e cabeças magnéticas e seus gabinetes normalmente não são blindados. Unidades em gabinetes metálicos provavelmente são seguras.
- Ventiladores portáteis normalmente contêm motores sem blindagem.
- Monitores usam transformadores e altas tensões. Muitos monitores retêm carga elétrica mesmo depois de terem sido desligados. Os monitores coloridos são os piores. Jamais guarde fitas sobre um monitor.
- Exposição prolongada à radiação terrestre afeta os dados em mídias magnéticas, limitando seu ciclo de vida. Todas as fitas se tornam ilegíveis após alguns anos. A maior parte da mídia é capaz de reter dados por três anos, porém, se pretende armazenar dados por mais tempo deveria usar mídia ótica ou regravar os dados.

As seções a seguir descrevem algumas das mídias que podem ser usadas para backups. As mídias são apresentadas, grosseiramente, em ordem de maior capacidade.

Muitas unidades de fita compactam dados antes de gravá-los em fita, permitindo que uma maior quantidade de dados seja armazenada superando a capacidade nominal da fita. Os fabricantes adoram apresentar as capacidades das fitas em termos de dados compactados: eles normalmente supõem de forma otimista uma razão de compactação de 2:1 ou mais. Nas seções a seguir, nós ignoramos a compactação e citamos o número de bytes em si que podem fisicamente ser armazenados em cada mídia.

A taxa de compactação presumida de uma unidade também afeta seu desempenho. Se uma unidade é capaz de gravar 1 MB/s em fita porém o fabricante supõe uma compactação de 2:1, a velocidade de gravação sobe, num passe de mágica, para 2 MB/s. Como acontece com os números de capacidade, também ignoramos esse "ganho" de velocidade.

Embora custo e capacidade de mídias sejam considerações importantes, também é importante considerar a velocidade. Mídias rápidas são mais agradáveis de lidar e elas oferecem uma maior flexibilidade na programação de backups.

Disquetes

Os disquetes são a forma mais inconveniente de armazenar backups. Eles são lentos e não armazenam grandes quantidades de dados. Embora os discos individualmente sejam baratos, eles armazenam tão pouco que são, na verdade, a mídia para backup mais cara. Os disquetes duram somente alguns anos; jamais os use para armazenamento de longo prazo. Por outro lado, as unidades de disquete são baratas e normalmente já vêm com o sistema.

Snow666

Superdisquetes

Os Zip drives da Iomega agora estão por toda a parte e normalmente são a mídia externa padrão para PCs domésticos. A capacidade de armazenamento destas unidades passou de 100 MB para 250 MB. As unidades são disponíveis com conectores paralelos, seriais, SCSI e USB.

A Imation comercializa um produto SuperDisk capaz de gravar tanto disquetes flexíveis tradicionais como mídia especial de 120 MB.

Embora esses produtos sejam úteis para troca de dados, os seus altos custos de mídia os tornam uma má opção para backups.

CD-R e CD-RW

As recentes diminuições no preço dos CDs somente graváveis e regraváveis os torna uma mídia para backups muito mais interessante do que foram alguns anos atrás. Ambas as formas armazenam 650 MB. As unidades que gravam estes CDs são disponíveis em tantas formas como são suas unidades de disco rígido preferidas: SCSI, IDE, paralelas, USB etc.

Os CDs são gravados com um laser por meio de um processo fotoquímico. Embora dados concretos tenham sido evasivos, acredita-se amplamente que os CDs têm uma vida útil de armazenamento mais longa do que a mídia magnética. Os CDs que podem ser gravados somente uma vez, conhecidos como CD-Rs, não são tão duráveis quanto CDs normais (estampados). O CD-R não é particularmente uma boa opção para backups normais, porém é bom para arquivamento de dados que talvez você possa querer recuperar daqui a um bom tempo.

A tecnologia DVD gravável está apenas começando. Antevemos a sua adoção como mídia para armazenamento de dados. Sua capacidade deve se situar entre 5GB a 10 GB.

Discos rígidos removíveis

Uma série de discos removíveis de alta capacidade tem sido lançada no mercado ao longo dos últimos anos. A Castlewood Industries possui um produto de 2,2 GB chamado Orb drive. Ele vem nas formas interna e externa Ultra SCSI, EIDE e USB. Para detalhes, vá ao site www.castlewood.com.

Um outro drive popular é o Iomega Jaz, que armazena 2 GB e diz ter uma taxa de transferência de 8,7 MB/s. O Jaz também clama uma vida útil de armazenamento de 10 anos, ao passo que a Orb define um nível mais realista de 5 anos. Maiores informações sobre o Jaz drive podem ser obtidas em www.iomega.com.

O mercado de unidades de disco removível está se tornando mais competitivo e os preços variam diariamente. A principal vantagem destes produtos é a velocidade: eles atingem taxas de transferência comparáveis às das unidades de disco normais. Eles são interessantes como dispositivos de backup para pequenos sistemas e máquinas domésticas, embora os discos em si sejam um tanto salgados.

Fitas-cartucho de 8 mm

Várias marcas de unidades de fita gravam em fitas de vídeo padrão de 8 mm (formato reduzido). As unidades são normalmente conhecidas como "Exabytes drives" em homenagem à primeira empresa que as produziu. O formato original armazenava 2 GB e os formatos mais recentes armazenam até 7 GB. A compactação de hardware embutida em algumas unidades elevam ainda mais essa capacidade.

O tamanho das fitas de 8 mm tornam o armazenamento fora da instalação muito conveniente. Originalmente, os mecanismos da unidade eram um tanto problemáticos no aspecto em que eles sairiam de alinhamento a cada 6-12 meses e exigiam uma manutenção custosa por parte do fabricante. Isso não é mais o caso.

As fitas de 8 mm vêm tanto no formato vídeo quanto dados. Alguns fabricantes insistem no formato dados e cancelam a sua garantia caso use mídia não aprovada. Achamos a forma vídeo razoavelmente aceitável caso ela não cancele a garantia. Esteja atento ao fato de que ambos os tipos de fitas são sensíveis ao calor.

Fitas DAT, cartuchos de 4mm

As unidades DAT (Digital Audio Tape) são dispositivos de varredura helicoidal que usam cartuchos de 4 mm. Embora estas unidades sejam normalmente conhecidas como unidades DAT, elas são na verdade unidades DDS (Digital Data Storage); a distinção exata não é importante. O formato original armazenava cerca de 2 GB, porém gerações sucessivas aumentaram significativamente a capacidade das DDSs. A geração atual (DDS-4) armazena até 20 GB.

As unidades DAT são rápidas na busca e transferem dados numa velocidade de até 2,5 MB/s (para DDS-4), tornando o sistema relativamente rápido. Sua alta capacidade possibilita que um backup completo seja feito sem a intervenção do operador em diversas instalações. As fitas de 4mm são compactas, reduzindo a necessidade de espaço

Snow666

para armazenamento e facilitando o armazenamento fora da instalação. As unidades de fitas DAT não possuem um histórico de problemas referentes a alinhamento.

Pelo menos um fabricante importante (a Sony) anunciou o fim de vida das DDSs, de modo que esta tecnologia talvez não seja a melhor opção para emprego numa empresa.

Fitas Travan

Merecem uma breve menção a próxima geração de fitas QIC, conhecida como Travan. As unidades Travan usam uma tecnologia de gravação linear e suportam mídia de 2,5 GB a 10 GB. As unidades são baratas, porém as fitas custam um pouco mais do que aquelas de outros sistemas de fita de alta capacidade, cerca de US$ 3/GB. O oba-oba do mercado reivindica uma taxa de transferência de cerca de 1 MB/s.

Muitos fabricantes produzem unidades Travan, porém o suporte para Linux é deficiente. Quando este livro foi escrito, as unidades para Linux disponíveis eram das marcas HP, Tandberg e Tecmar.

DLT

Digital Linear Tapes é uma mídia de backup popular. Estas unidades são confiáveis, de preço razoável e alta capacidade. Elas evoluíram a partir das unidades de fita-cartucho TK-50 e TK-70 da DEC, que eram periféricos populares na DEC VAXstation. A primeira geração das unidades DLT era capaz de ler as antigas fitas TK-70. A DEC vendia a tecnologia à Quantum, que popularizou as unidades aumentando suas velocidades e capacidades e diminuindo os seus preços. Infelizmente, os mecanismos de transporte da fita são basicamente os mesmos daqueles do projeto TK-50 original que era popular nos anos 80. Elas precisam de uma remodelagem.

As fitas DLT armazenam uma boa quantidade de dados, até 40 GB. As taxas de transferência estão na casa dos 6 MB/s. Os fabricantes anunciam que as fitas durarão de 20 a 30 anos; isto é, se o hardware para lê-las ainda existir nessa época. Quantas unidades de fita de 9 trilhas ainda estão funcionando e estão ativas nos dias de hoje?

O inconveniente do DLT é o preço da mídia, que gira em torno de US$ 65 por fita, embora os preços estejam diminuindo. Para uma universidade, essa é uma despesa muito alta; para uma empresa de investimentos da Wall Street, esse custo pode ser razoável.

AIT

Advanced Intelligent Tape é um produto de 8 mm proprietário da Sony. Em 1996, a Sony rompeu o seu relacionamento com a Exabyte e introduziu o AIT-1, um dispositivo de varredura helicoidal de 8 mm com o dobro da capacidade das unidades de 8 mm da Exabyte. Desde o lançamento do produto original, a Sony introduziu versões de maior capacidade do que o AIT-1, seguindo os AIT-2 e AIT-3 (em 2001).

As fitas AME (Advanced Metal Evaporated) usadas em unidades AIT possuem uma vida útil longa. Elas também contêm uma EEPROM embutida que confere uma certa inteligência à própria fita. Entretanto é necessário suporte de software para fazer uso efetivo da EEPROM. O AIT-2 clama taxas de transferência nativas de 6 MB/s com uma capacidade de mídia de 50 GB. O AIT-3 clama o dobro desses números, com uma taxa de transferência nativa de 12 MB/s e uma capacidade de mídia de 100 GB. Os preços das unidades e das fitas equivalem grosseiramente às da DLT.

Mammoth

A Mammoth da Exabyte é uma versão aperfeiçoada do sistema de fitas de 8 mm. A Exabyte optou por desenvolver e fabricar seu próprio mecanismo após sua dissidência com a Sony. Os produtos Mammoth ainda dependem da Sony por causa da mídia AME, porém as fitas não contêm um chip de memória.

As primeiras gerações da linha Mammoth apresentavam problemas de confiabilidade. Contudo, a Exabyte ouviu os seus clientes e aperfeiçoou os produtos. Hoje o fabricante clama uma taxa de falhas de apenas 1%.

As discórdias originais entre a Exabyte e a Sony se deviam à baixa qualidade de fabricação da Sony. A Sony utilizava somente uma linha de produção tanto para produtos ao consumidor como para unidades de fita, de modo que a Exabyte era aos olhos da Sony idêntica a uma câmera de vídeo. Porém, a Sony parece estar à frente agora.

Jukeboxes, empilhadores e bibliotecas de fitas

Com o baixo custo dos discos atualmente, a maioria das instalações tem tanto espaço em disco que um backup completo exigiria várias fitas, até mesmo fitas de 20 GB de capacidade. Uma solução para essas instalações é um empilhador, uma jukebox ou uma biblioteca de fitas.

Um empilhador é um trocador de fitas simples usado com uma unidade de fita padrão. Ele possui um depósito alimentador no qual carregamos fitas; ele descarrega fitas completas à medida que são ejetadas da unidade e as substitui por fitas em branco alojadas no depósito alimentador. A maioria dos empilhadores tem capacidade para até dez fitas.

Snow666

Uma jukebox é um dispositivo de hardware capaz de trocar automaticamente mídia removível num número limitado de unidades, de maneira muito parecida com aqueles trocadores de disco de vinil antigos. As jukeboxes são encontradas em vários tipos de mídia, inclusive DAT, DLT, AIT e CD. As jukeboxes normalmente vêm com software para backup especial que sabe como lidar com o trocador. A Storage Technology e a Sony são dois fabricantes desse tipo de produto.

As bibliotecas de fitas são uma solução para backup para grandes conjuntos de dados — terabytes, normalmente. Estas são mecanismos do tamanho de um armário com varias unidades de fita (ou CDs) e um braço mecânico que retira e coloca mídia nas diversas prateleiras de sua biblioteca. Como é possível imaginar, elas são bem caras em termos de aquisição e de manutenção e exigem alimentação, espaço e condições de climatização especiais. A maioria dos compradores de bibliotecas de fitas também adquire um contrato de operações do fabricante para otimizar e comandar o equipamento. Obviamente, as bibliotecas possuem um componente de software que é o que realmente comanda o dispositivo. A Storage Technology é um fabricante de ponta de bibliotecas de fitas.

Discos rígidos

Seríamos negligentes se não mencionássemos a diminuição do custo das unidades de disco rígido como uma razão para considerar a possibilidade de backup de disco para disco. Embora sugerimos que você não duplique um disco em outro dentro da mesma máquina física, os discos rígidos podem ser uma boa solução e de custo baixo para armazenamento através de uma rede.

Um problema óbvio é que o espaço em armazenamento em disco rígido é finito e tem eventualmente de ser reutilizado. Porém, backups de disco para disco é uma excelente maneira de se proteger contra o apagamento acidental de arquivos. Se você mantém uma imagem de disco defasada de um dia num local bem conhecido que é compartilhado através de NFS ou CIFS, os usuários poderão se recuperar de seus próprios erros sem ter de envolver o administrador do sistema.

Síntese dos tipos de mídia

Ufa! Há realmente um monte de possibilidades. A Tabela 10.1 sintetiza as características discutidas nas seções anteriores.

Tabela 10.1 Comparação entre mídias para backup

Mídia	Capacidade[a]	Veloc.[a]	Drive	Mídia	Custo/GB	Reuso?	Aleatório?
Disquete	2.8MB	< 100 KB/s	US$15	25¢	US$91,43	Sim	Sim
SuperDisk ou Zip Disk	120MB	1.1 MB/s[b]	US$200	US$8	US$68,27	Sim	Sim
Zip 250	250MB	900 KB/s	US$200	US$15	US$61,44	Sim	Sim
CD-R	650MB	2.4 MB/s	US$200	75¢	US$1,18	Não	Sim
CD-RW	650MB	2.4 MB/s	US$200	US$2	US$3,15	Sim	Sim
Jaz	2GB	7.4 MB/s	US$350	US$100	US$50,00	Sim	Sim
Orb	2.2GB	12.2 MB/s[b]	US$200	US$40	US$18,18	Sim	Sim
Exabyte (8mm)	7GB	1 MB/s	US$1.200	US$8	US$1,14	Sim	Não
Travan	10GB	1 MB/s	US$200	US$34	US$3,40	Sim	Não
DDS-4 (4mm)	20GB	2.5 MB/s	US$1.000	US$30	US$1,50	Sim	Não
DLT (1/2 in.)	40GB	6 MB/s	US$4.000	US$60	US$1,50	Sim	Não
AIT-2 (8mm)	50GB	6 MB/s	US$3.500	US$95	US$1,90	Sim	Não
AIT-3 (8mm)	100GB	12 MB/s	US$5.000	US$170	US$1,70	Sim	Não
Mammoth-2	60GB	12 MB/s	US$3.500	US$80	US$1,33	Sim	Não

a. Velocidade e capacidade descompactadas.
b. Pico máximo da taxa de transferência; o fabricante não revela o desempenho médio verdadeiro.

W. Curtis Preston compilou uma excelente lista de referência de dispositivos para backup classificados por fabricante. Ela se encontra disponível em www.backupcentral.com/hardware-drives.html.

Snow666

O que comprar

Ao comprar um sistema de backup, você provavelmente obterá exatamente o que vimos na Tabela 10.1. Todas as mídias funcionam razoavelmente bem e entre as tecnologias próximas em preço, geralmente não há uma forte razão para optar por um ou por outro. Compre um sistema que atenda às suas especificações e caiba no seu orçamento.

As unidades DAT e Exabyte são excelentes soluções para pequenos grupos de trabalhos e para máquinas individuais com grande capacidade de armazenamento. Os custos iniciais são relativamente modestos, as mídias são encontradas em abundância e vários fabricantes estão usando os padrões de cada um deles. Ambos os sistemas são rápidos o bastante para fazer o backup de uma quantidade grande de dados numa quantidade finita de tempo.

DLT, AIT e Mammoth-2 são grosseiramente comparáveis. Não há um ganhador claro entre os três e mesmo se houvesse, a situação sem dúvida nenhuma mudaria em poucos meses à medida que novas versões dos formatos fossem empregadas. Todos esses formatos funcionam bem, e todos eles atingem o mesmo mercado: universidades e ambientes corporativos que precisam de hardware sério para backup (que, por sinal, envolve praticamente todas as universidades e ambientes corporativos).

Nas seções seguintes, usamos o termo genérico "fita" para nos referirmos à mídia escolhida para backups. Exemplos dos comandos de backup são feitos em termos de dispositivos de fitas.

10.3 Configurando um Regime de Backup Incremental via Dump

Os comandos **dump** e **restore** são a maneira mais comum de criarmos e restaurarmos informações de backup. Esses programas existem há um longo tempo e seus comportamentos são bem conhecidos. Na maioria das instalações, **dump** e **restore** são comandos subjacentes usados por software de backup automatizado.

Talvez seja necessário instalar explicitamente **dump** e **restore** em seus sistemas Linux, dependendo de as opções selecionadas durante a instalação original. Um arquivo **rpm** (Red Hat Package Manager) se encontra disponível para fácil instalação tanto pelo Red Hat como pelo SuSE, e no Debian, um pacote **deb**, contendo **dump** e **restore**, pode ser instalado com o apt-get. No Linux, nada é linkado estaticamente, de modo que precisamos de bibliotecas compartilhadas em **/lib** para fazer alguma coisa útil. A linkagem estática facilita a recuperação após um desastre, pois **restore** é então completamente autocontido.

Fazendo o dump de sistemas de arquivos

dump cria uma lista de arquivos que foram modificados desde um dump anterior, depois empacota estes arquivos num único grande arquivo para arquivamento em um dispositivo externo. **dump** apresenta várias vantagens em relação à maioria dos demais utilitários descritos neste capítulo:

- Backups podem abranger várias fitas.
- Arquivos de qualquer tipo (até mesmo dispositivos) podem ser colocados em backup e restaurados.
- Permissões, propriedades e horários de modificação são preservados.
- Arquivos com buracos são tratados corretamente[4].
- Backups podem ser realizados incrementalmente (apenas com os arquivos modificados recentemente gravados na fita).

A versão GNU do **tar** usada no Linux fornece todos esses recursos também. Porém, a manipulação por parte de **dump** de backups incrementais (discutidos posteriormente) é um pouco mais sofisticada do que a do **tar**. Talvez você ache esse poder adicional útil caso suas necessidades sejam complexas.

Na realidade, a razão mais forte para optarmos por **dump** em relação a **tar** num ambiente Linux não tem nada a ver com o Linux em si. Infelizmente, a versão de **tar** distribuída com a maioria das distribuições do UNIX não é capaz de realizar os maravilhosos recursos listados acima. Se você tiver de suportar backups tanto para Linux como para UNIX, **dump** é a melhor opção. É o único comando que trata dessas questões de forma relativamente consistente nessas duas plataformas, permitindo que você seja um expert em um comando em vez de ser familiarizado com dois. Se tiver sorte o bastante para estar num ambiente Linux completamente homogêneo, escolha o seu favorito. **dump** tem menos recheio, porém **tar** tem um excelente sabor!

---

4. Buracos são blocos que jamais contiveram dados. Se você abrir um arquivo, gravar um byte, pesquisar 1 MB no arquivo, depois gravar um outro byte, o arquivo "esparso" resultante ocupará somente dois blocos de disco muito embora seu tamanho lógico seja muito maior. Os arquivos criados pelos utilitários Berkeley **db** ou **ndbm** contêm muitos buracos.

www.snow666.blogspot.com
Snow666

O comando **dump** compreende o layout de sistemas de arquivos brutos e lê as tabelas de inodes de um sistema de arquivos diretamente para decidir quais arquivos devem ser colocados em backup. Esse conhecimento do sistema de arquivos permite a **dump** ser muito eficiente, porém ele também impõe algumas limitações. [5]

A primeira limitação é que todo sistema de arquivos deve ter o seu backup feito individualmente. Se tivermos um disco que é particionado, temos de fazer o dump de cada partição separadamente. A outra limitação é que somente sistemas de arquivos na máquina local podem ter o dump feito; não podemos fazer o dump de um sistema de arquivos NFS montado de uma máquina remota. Porém, podemos efetuar o dump de um sistema de arquivos local para uma unidade de fita remota com o irmão gêmeo de **dump**, o **rdump**.

O recurso mais importante de **dump** é o seu suporte ao conceito de backup "incremental". Embora seja possível fazer o backup do sistema inteiro todo dia, normalmente não é prático. Dumps incrementais possibilitam que façamos o backup somente de arquivos que foram alterados desde o último backup.

Ao fazermos um dump, atribuímos a ele um nível de backup, que é um inteiro de 0 a 9. Um dump nível N faz o backup de todos os arquivos que foram modificados desde o último dump de nível menor do que N. Um backup nível 0 coloca todo o sistema de arquivos na fita. Com um sistema de backup incremental, talvez tenhamos de restaurar arquivos de vários conjuntos de fitas de backup para recolocar um sistema de arquivos no estado em que se encontrava durante o último backup. [6]

Um outro recurso interessante de **dump** é que ele não se importa com o comprimento de nomes de arquivos. As hierarquias podem ser arbitrariamente profundas, e nomes longos são manipulados corretamente.

O primeiro argumento para dump tem de ser o nível de dump incremental. **dump** usa o arquivo **/etc/dumpdates** para determinar até que nível se deve voltar num dump incremental. O flag **-u** faz com que **dump** atualize automaticamente **/etc/dumpdates** quando o dump é completado. São gravados a data, nível de dump e o nome de sistema de arquivos. Se jamais usarmos o flag **-u**, todos os dumps se tornam nível 0, pois não há nenhum registro de se ter aplicado dump ao sistema de arquivos. Se você modificar o nome de um sistema de arquivos, poderá editar manualmente o arquivo **/etc/dumpdates**.

Veja a seção 12.5 para obter maiores informações sobre números de dispositivos.

dump envia sua saída para algum dispositivo-padrão, normalmente a unidade de fita primária. Se quiser usar um dispositivo diferente, use o flag **-f** para informar **dump** para enviar sua saída para um outro lugar. Se tiver colocando vários dumps numa única fita, certifique-se de especificar um dispositivo de fita não rebobinável (um arquivo de dispositivo que não faça com que a fita seja rebobinada quando é fechada — a maioria das unidades de fita possui ambos, uma entrada de dispositivo padrão e uma entrada de dispositivo não rebobinável). Leia a página do manual referente a dispositivos de fita para determinar o nome exato do arquivo de dispositivos apropriado. [7] O Linux normalmente usa **/dev/st0** para o dispositivo rebobinável e **/dev/nst0** para o dispositivo não rebobinável.

Se, acidentalmente, optar pelo dispositivo rebobinável, acabará salvando apenas o último sistema de arquivos ao qual foi aplicado **dump**. Já que **dump** não possui a mínima idéia de onde a fita se encontra posicionada, este engano não provoca erros. A situação somente se torna aparente quando você tenta restaurar arquivos.

Ao usarmos **rdump** para fazer o dump para um sistema remoto, especificamos a identidade da unidade de fita remota como *hostname:device*; por exemplo,

```
# rdump -0u -f anchor:/dev/nst0 /spare
```

A permissão de acesso a unidades de fita remotas é controlada pelo mecanismo **.rhosts**. Recomendamos que você use em vez disso um túnel SSH. Veja o Capítulo 21 para obter maiores informações.

No passado tínhamos de informar a **dump** o comprimento exato das fitas para que pudesse parar a gravação antes de atingir o final da fita. A maioria das unidades de fitas modernas é capaz de informar quando atinge o final de uma fita e pode relatar este fato de volta para **dump**, que, a seguir, rebobina e ejeta a fita atual e solicita uma nova. Já que a variabilidade de compactação de hardware torna o "comprimento virtual" de cada fita um tanto indeterminado, é sempre melhor depender da indicação do final de fita caso o nosso hardware o suporte. Caso contrário, podemos especificar o comprimento da fita em kilobytes com a opção **-B**.

---

5. **dump** requer acesso a partições de discos brutas. Qualquer um com permissão para realizar dumps pode ler todos os arquivos no sistema com um pouco de trabalho.
6. Na realidade, **dump** não controla os arquivos que foram eliminados. Se restaurarmos de backups incrementais, os arquivos eliminados serão recriados.
7. Todas as entradas para uma unidade de fita usam o mesmo número de dispositivo principal. O número de dispositivo secundário informa o driver sobre comportamentos especiais (rebobinagem, swapping de bytes etc.).

Snow666

O flag **-B** também pode ser útil em situações funestas quando somos forçados a usar um cartucho que apresenta um defeito de mídia mais para o final da fita. Suponhamos, por exemplo, que queiramos fazer um dump nível 5 de **/work** numa unidade DDS-1 (DAT) cuja capacidade nativa é de 1 GB. São 3 horas da manhã de domingo e no seu armário de suprimentos não tem mais nada a não ser uma pilha de fitas DAT defeituosas e meio saquinho de Cheetos.[8] A primeira fita escolhida está rotulada como tendo uma falha aproximadamente na marca 850 MB. Nessa situação, poderíamos usar a opção **-B** para informar **dump** de que o comprimento da fita é ligeiramente inferior do que realmente é.

```
# dump -5u -B 800000 -f /dev/st0 /work
DUMP: Date of this level 5 dump: Wed May 8 16:59:45 2002
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/hda2 (/work) to /dev/st0
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 18750003 tape blocks on 1.51 tape(s).
...
```

Os flags **-5u** são seguidos pelos flags **-B** (comprimento da fita em KB:800 MB) e **-f** (dispositivo de fita: **/dev/st0**). O último argumento, que é obrigatório, é o nome do sistema de arquivos a ser colocado em dump (**/work**). A última linha da saída acima verifica se **dump** trocará fitas por conta própria quando atingir cerca de 2/3 do caminho do processo de backup, evitando o defeito presente na mídia.

Seqüências de dump

Pelo fato de os níveis de dump serem arbitrários (eles possuem significado somente em relação a outros níveis), os dumps podem ser executados em várias escalas. A escala adequada para o seu caso depende da:

- Atividade de seus sistemas de arquivos
- Capacidade de seu dispositivo de dump
- Quantidade de redundância desejada
- Número de fitas que deseja adquirir

Na época em que se usava muitas fitas para se fazer o backup de um sistema de arquivos, seqüências de dump complicadas eram úteis para minimizar o número de fitas consumidas pelo backup diário. À medida que a capacidade das fitas foi aumentando, tornou-se menos útil fazer-se distinções entre níveis de dump.

Pelo fato de a maioria dos arquivos jamais mudar, até mesmo a programação incremental mais simples elimina muitos arquivos dos dumps diários. À medida que adicionamos níveis à nossa programação de dump, dividimos os relativamente poucos arquivos em segmentos cada vez menores.

Uma programação de dump complexa gera os seguintes benefícios:

- Podemos fazer backup de dados com maior freqüência, limitando o potencial de perdas.
- Podemos usar menos fitas diárias (ou fazer caber tudo numa única fita).
- Podemos manter várias cópias de cada fita, para proteção contra erros nas fitas.

Em geral, a maneira de selecionarmos a seqüência é determinar nossas necessidades em cada uma dessas áreas. Dadas essas restrições, podemos elaborar um cronograma no nível apropriado de sofisticação. Descrevemos uma série de seqüências possíveis e a motivação por trás delas. Uma delas deve ser adequada ao seu caso — caso contrário, suas necessidades talvez ditem uma programação completamente diferente.

Uma programação simples

Se a quantidade total de espaço em disco for menor do que a capacidade de seu dispositivo de fita, poderá usar um esquema de dump completamente trivial. Faça dumps nível 0 de todos os sistemas de arquivos a cada dia. Reutilize

8. Um sintoma da Síndrome de Personalidade do Administrador de Sistemas é o habitual empilhamento de computadores e acessórios parcialmente operativos. É melhor deixar identificado o que está errado com aquilo que você joga na pilha para poder opinar sobre sua utilidade numa emergência. Nota: se você se vir retirando a solda de chips de placas de circuito nesta pilha "cemitério" e os transplantando para máquinas para mantê-las funcionando "somente mais um dia", é hora de procurar ajuda psiquiátrica. Acredite em nós, todos já passamos por isso.

Snow666

um grupo de fitas, porém, a cada N dias (onde N é determinado pelas necessidades de sua instalação), mantenha a fita para sempre. Este sistema lhe custará

```
(365/N) * (preço da fita)
```

por ano. Não reutilize exatamente a mesma fita para o dump de cada noite. É melhor fazer o rodízio entre um conjunto de fitas para que, mesmo que um dump possa vir a estar defeituoso, ainda possamos retornar ao estado da noite anterior.

Esta programação fornece redundância massiva e torna a recuperação de dados muito fácil. É uma boa solução para uma instalação com muito dinheiro, porém tempo (ou habilidade) de operador escasso. Sob uma perspectiva de conveniência e de segurança, esta programação é a ideal. Não se afaste dela sem ter uma razão específica (por exemplo, conservar fitas ou trabalho).

Uma programação moderada

Uma programação mais razoável para instalações é atribuir uma fita a cada dia da semana, cada semana do mês (você precisará de 5) e cada mês do ano. Todo dia, faça um dump nível 9 para a fita diária. A cada semana, faça um dump nível 5 para a fita semanal. E a cada mês, faça um dump nível 3 para a fita mensal. Faça um dump nível 0 toda vez que os incrementos se tornarem muito grandes para caberem numa fita, o que é mais provável de acontecer numa fita mensal. Faça um dump nível 0 pelo menos uma vez por ano.

A escolha dos níveis 3, 5 e 9 é arbitrária. Você poderia usar os níveis 1, 2 e 3 com o mesmo efeito. Porém, os intervalos entre os níveis de dump nos dão alguma folga caso decidamos posteriormente adicionar um outro nível de dump.

Esta programação requer 24 fitas além de tantas fitas quantas forem necessárias para executar os dumps nível 0. Embora ela não exija tantas fitas assim, ela também não é capaz de fornecer muita redundância.

10.4 Restaurando Dumps com Restore

O programa que extrai dados de fitas gravadas com **dump** é chamado **restore**. Discutiremos primeiramente a restauração de arquivos individuais (ou um pequeno conjunto de arquivos), depois explicaremos como restaurar sistemas de arquivos inteiros.

Restaurando arquivos individuais

A primeira providência a tomar quando você é notificado da perda de um arquivo é determinar que fitas contêm versões do arquivo. Os usuários normalmente querem a versão mais recente de um arquivo, porém isto nem sempre é o caso. Por exemplo, um usuário que perde um arquivo copiando inadvertidamente um outro arquivo por cima dele gostaria de ter a versão que existia antes da ocorrência deste incidente. Ajuda se você puder intimidar os usuários a informarem-no não somente quais arquivos estão faltando, como também quando eles foram perdidos e quando foram modificados pela última vez.

Caso não mantenha catálogos on-line, terá de montar fitas e tentar repetidamente restaurar os arquivos faltantes até encontrar a fita correta. Se o usuário se lembrar de quando os arquivos foram modificados pela última vez, talvez você possa fazer uma estimativa mais provável de em quais fitas eles poderiam estar.

Após determinar de que fitas fazer a extração, crie um diretório temporário e mude pare ele (**cd**) como, por exemplo, **/var/restore** onde uma hierarquia de diretórios maior pode ser criada; a maioria das versões de **restore** precisa criar todos os diretórios que levam a um determinado arquivo antes de este arquivo poder ser restaurado. Não use **/tmp** — seu trabalho poderia ser eliminado caso a máquina trave e reinicialize antes de os dados restaurados terem sido deslocados para suas posições originais.

O comando **restore** possui várias opções. As mais úteis são **-i** para restaurações interativas de arquivos e diretórios individuais e **-r** para uma restauração completa de um sistema de arquivos inteiro. Talvez você também precise de **-x**, que solicita uma restauração não interativa de arquivos especificados — tome cuidado para não sobregravar arquivos existentes.

restore -i lê a tabela de conteúdo da fita e depois nos permite navegar através dela como se fosse uma árvore de diretórios comum, usando comandos chamados **ls**, **cd** e **pwd**. Marcamos os arquivos que queremos restaurar por meio do comando **add**. Quando tiver terminado de selecionar os arquivos, digite **extract** para extrair os arquivos da fita.

*Veja a seção 10.7 para obter uma descrição de **mt**.*

Caso tenha colocado vários arquivos em uma única fita, terá de usar o comando **mt** para posicionar a fita no arquivo de dump correto antes de executar **restore**. Lembre-se de usar o dispositivo não rebobinável!

Snow666

Para, por exemplo, restaurar o arquivo **/users/janet/iamlost** de uma unidade de fita remota, talvez você tenha de emitir os seguintes comandos. Suponhamos que você tenha localizado a fita correta, a tenha montado em **tapehost:/dev/nst0** e tenha determinado que o sistema de arquivos contendo o diretório de janet seja o quarto na fita.

```
# mkdir /var/restore
# cd /var/restore
# rsh[9] tapehost mt -f /dev/nst0 fsf 3
# rrestore -i -f tapehost:/dev/nst0
restore> ls
.:
janet/ garth/ lost+found/ lynda/
restore> cd janet
restore> ls
afile bfile cfile iamlost
restore> add iamlost
restore> ls[10]
afile bfile cfile iamlost*
restore> extract
You have not read any volumes yet.
Unless you know which volume your files are on you should
start with the last volume and work towards the first.
Specify next volume #: 1
set owner/mode for '.'? [yn] n
```

Os volumes (fitas) são numerados iniciando em 1, não em 0, de modo que para um dump que cabe numa única fita especificamos 1. Quando **restore** pergunta se você quer configurar o proprietário e o modo para ".", ele está perguntando se ele deve configurar o diretório atual para coincidir com a raiz da fita. A menos que você esteja restaurando um sistema de arquivos inteiro, provavelmente não vai querer fazer isso.

Assim que **restore** tiver terminado, você precisará dar o arquivo a Janet:

```
# cd /var/restore
# ls janet
iamlost
# ls ~janet
afile bfile cfile
# cp -p janet/iamlost ~janet/iamlost.restored
# chown janet ~janet/iamlost.restored
# rm -rf /var/restore
# mail janet
Your file iamlost has been restored as requested and has
been placed in /users/janet/iamlost.restored.

Your name, Humble System Administrator
```

Alguns administradores preferem restaurar arquivos num diretório especial, permitindo aos usuários copiarem seus arquivos manualmente. Nesse esquema, o administrador deve proteger a privacidade dos arquivos restaurados verificando quem são os seus proprietários e respectivas permissões. Se optar por usar um sistema como esse, lembre-se de limpar o diretório de vez em quando.

Caso crie um backup com **rdump** e seja incapaz de restaurar arquivos dele com o comando **restore**, tente executar **rrestore**. Para minimizar as chances de problemas, use algum host para ler a fita como foi usado para gravá-la.

restore -i normalmente é a maneira mais fácil de restaurar alguns arquivos ou diretórios de um dump. Porém, ele não funcionará caso o dispositivo de fita não possa ser movido para trás um registro por vez (um problema que ocorre com alguns dispositivos de 8 mm). Se **restore -i** falhar, tente **restore -x** antes de pular pela janela. **restore -x** requer que especifiquemos o caminho completo do arquivo que queremos restaurar (relativo à raiz do dump) na linha de comandos. A seqüência de comandos a seguir repete o exemplo anterior, porém com **restore -x**:

```
# mkdir /var/restore
# cd /var/restore
# rsh tapehost mt -f /dev/nst0 fsf 3
# rrestore -x -f tapehost:/dev/nst0 ./janet/iamlost
```

9. Você poderia usar em vez disso o comando **ssh** para maior segurança.
10. O asterisco próximo a **iamlost** indica que ele foi marcado para extração.

www.snow666.blogspot.com
Snow666

Restaurando sistemas de arquivos inteiros

Com sorte, você jamais terá de restaurar um sistema de arquivos inteiro após uma falha no sistema. Porém, esta situação ocorre eventualmente. Antes de tentar restaurar o sistema de arquivos você tem de, em primeiro lugar, ter certeza de qual problema fez com que o sistema de arquivos fosse destruído. É decepcionante gastar horas e horas rebobinando fitas somente para perder o sistema de arquivos novamente.

Antes de iniciar uma restauração completa, você tem de criar e montar o sistema de arquivos de destino. Consulte o Capítulo 8, *Adicionando um Disco*, para obter maiores informações sobre como preparar o sistema de arquivos.

Para iniciar a restauração, dê o comando **cd** para o ponto de montagem do novo sistema de arquivos, coloque a primeira fita do dump nível 0 mais recente e digite **restore -r**.

restore irá lhe solicitar cada fita no dump. Depois de o nível 0 ter sido restaurado, monte e restaure os dumps incrementais. Restaure dumps incrementais na ordem em que eles foram criados. Devido à redundância entre dumps, talvez não seja necessário restaurar todos os incrementais. Eis o algoritmo para determinar que dumps restaurar:

Etapa 1: Restaure o nível 0 de dump mais recente.
Etapa 2: Restaure o dump de nível mais baixo após o dump que acabou de restaurar. Se forem feitos vários dumps neste nível, restaure aquele mais recente.
Etapa 3: Se este foi o último dump que foi feito, está terminado o processo.
Etapa 4: Caso contrário, retorne para a etapa 2.

Eis alguns exemplos de seqüências de dump. Você precisaria restaurar somente os níveis indicados em negrito.

```
0 0 0 0 0 0
0 5 5 5 5
0 3 2 5 4 5
0 9 9 5 9 9 3 9 9 5 9 9
0 3 5 9 3 5 9
```

*Veja o Capítulo 8 para obter maiores informações sobre os comandos **mke2fs** e **mount**.*

Vamos dar uma olhada na seqüência de comandos completa. Se o dump mais recente fosse o primeiro mensal após o nível 0 anual na programação "moderada" da página 132, os comandos para restaurar **/home**, residentes no dispositivo físico **/dev/sda1**, ficariam da seguinte maneira (os nomes do dispositivo são dependentes do hardware):

```
# /etc/mke2fs /dev/sda1 QUANTUM_PD1050S
# /etc/mount /dev/sda1 /home
# cd /home
/* Monte a primeira fita do dump nível 0 de /home. */
# restore -r
/* Monte as fitas solicitadas por restore. */
/* Monte a primeira fita do dump mensal de nível 3. */
# restore -r
```

Se tivéssemos vários sistemas de arquivos numa única fita de dump, teríamos de usar o comando **mt** para avançar para o sistema de arquivos correto antes de executar cada restauração. Essa seqüência restauraria o sistema de arquivos ao estado em que ele se encontrava quando o dump nível 5 foi feito, exceto que todos os arquivos eliminados seriam ressuscitados de forma macabra. Esse problema pode ser especialmente ruim quando você estiver restaurando um sistema de arquivos ativo ou estiver restaurando para um disco que está praticamente cheio. É bem possível para um **restore** falhar devido ao sistema de arquivos ter sido preenchido com arquivos fantasmas.

10.5 Realizando Dump e Restore para Atualizações

Ao executarmos uma atualização de sistema operacional temos de fazer o backup de todos os sistemas de arquivos com um dump nível 0 e, possivelmente, restaurá-los. A restauração é necessária somente se o novo sistema operacional usar um formato de sistema de arquivos distinto ou se você modificar o particionamento dos discos. Entretanto, você tem de fazer backups como uma garantia contra quaisquer problemas que possam vir a ocorrer durante a atualização. Um conjunto completo de backups também lhe dá a opção de reinstalar o sistema operacional antigo caso a nova versão não demonstre ser satisfatória.

Certifique-se de fazer backups e restaurar quaisquer arquivos específicos de sistema que se encontram em **/** ou **/usr**, como **/etc/passwd**, **/etc/shadow** ou **/usr/local**. A organização de diretórios do Linux mistura arquivos locais com arquivos distribuídos pelos fornecedores de sistemas, tornando difícil escolher suas personalizações locais.

www.snow666.blogspot.com
Snow666

Você deveria fazer também um conjunto completo de dumps nível 0 imediatamente após uma atualização. A maioria dos procedimentos de atualização dos fornecedores configura as datas de modificação de arquivos de sistema para o horário em que eles foram masterizados, e não para o horário atual. Portanto, dumps incrementais feitos em relação ao nível 0 anterior à atualização não serão suficientes para restaurar o seu sistema ao estado pós-atualização no caso de uma pane.

10.6 Usando Outros Programas de Arquivamento

dump não é o único programa que podemos usar para arquivamento de arquivos em fitas; porém, normalmente ele é a maneira mais eficiente de fazer o backup de um sistema inteiro. **tar**, **cpio** e **dd** também podem mover arquivos de uma mídia para outra.

tar: arquivos de pacotes

tar lê vários arquivos ou diretórios e os empacota num único arquivo, normalmente um arquivo em fita. **tar** é uma maneira útil de fazer backup de quaisquer arquivos cuja recuperação a curto prazo é prevista. Por exemplo, se um usuário estará ausente por seis meses e o sistema se encontra com pouco espaço em disco, podemos usar **tar** para colocar os arquivos do usuário numa fita e depois eliminá-los do disco.

tar também é útil para mover árvores de diretórios de um lugar para outro, especialmente se estivermos copiando arquivos como root (**tar** preserva as informações de propriedade). Por exemplo,

```
tar -cf - fromdir | ( cd todir ; tar -xfp - )
```

cria uma cópia da árvore de diretórios *fromdir* para *todir*. Evite usar ".." no argumento *todir* já que links simbólicos e automontadores podem torná-lo algo diferente daquilo que esperamos. Já caímos nessa várias vezes.

Como padrão **tar** não segue links simbólicos, porém podemos informá-lo para proceder desta forma. **tar** também pode ser informado para incluir somente arquivos que foram modificados desde uma dada data, o que é útil para criar seu próprio esquema de backup incremental. Consulte a página de manual referente a **tar** para esse e outros recursos interessantes.

Um problema com algumas versões não-Linux de **tar** é que os nomes de caminhos são limitados por padrão a 100 caracteres. Essa restrição impede que o **tar** arquive hierarquias mais profundas. Caso esteja criando arquivos tar em seus sistemas Linux e trocando-os com outros, lembre-se de que pessoas com o **tar** padrão talvez não sejam capazes de ler as fitas ou arquivos que você criou.

A opção **-b** de **tar** nos permite especificar um "fator de bloqueio" para ser adotado ao gravar uma fita. O fator de bloqueio é especificado em blocos de 512 bytes; ele determina que quantidade de dados o comando **tar** coloca em buffers internos antes de executar uma operação de gravação. Alguns dispositivos DAT não funcionam corretamente a menos que o fator de bloqueio seja configurado para um valor especial, porém outras unidades não requerem este tipo de configuração.

Em alguns sistemas, determinados fatores de bloqueio podem levar a um desempenho melhor do que outros. O fator de bloqueio ótimo varia muito, dependendo do hardware do computador e da unidade de fita. Em muitos casos, não perceberemos qualquer diferença em termos de velocidade. Quando em dúvida, tente um fator de bloqueio 20.

O **tar** do Linux expande buracos em arquivos a menos que usemos a opção **-S** ao criarmos o arquivo original. O **tar** do Linux é relativamente intolerante a erros de fita.

cpio: utilitário de arquivamento dos primórdios

cpio é similar a **tar** em funcionalidade. Ele data dos primórdios e raramente é usado hoje. Como **tar**, **cpio** pode ser usado para mover árvores de diretórios. O comando

```
find fromdir -depth -print | cpio -pdm todir
```

faria uma cópia da árvore de diretórios *fromdir* para *todir*. A versão GNU de **cpio** usada no Linux possibilita vários volumes de fitas, porém a maioria delas não permite. Somente o superusuário consegue copiar arquivos especiais. Mesmo que esteja familiarizado com **cpio** de um outro sistema, recomendamos que você reveja cuidadosamente a página de manual referente ao comando, já que as opções variam muito de sistema para sistema.

dd: rodar bits

dd é um programa para cópia e conversão de arquivos. A menos que lhe seja dito para fazer algum tipo de conversão, **dd** simplesmente copia seu arquivo de entrada para o seu arquivo de saída. Se um usuário lhe traz uma fita que foi escrita em algum sistema não-Linux, pode ser que **dd** seja o único modo de lê-lo.

Snow666

Um emprego histórico para **dd** era o de criar uma cópia de um sistema de arquivos inteiro. Porém, uma melhor opção nos dias de hoje é aplicar **mke2fs** ao sistema de arquivos de destino e depois executar **dump** acoplado a **restore**. **dd** pode algumas vezes danificar informações de particionamento se usado incorretamente. Ele pode somente copiar sistemas de arquivos entre partições cujos tamanhos são exatamente iguais.

dd também pode ser usado para fazer uma cópia de uma fita magnética. Com duas unidades de fita, digamos, **/dev/st0** e **/dev/st1**, deveríamos usar o comando

```
% dd if=/dev/st0 of=/dev/st1 cbs=16b
```

Com uma unidade (**/dev/st0**), você deveria usar a seguinte seqüência:

```
% dd if=/dev/st0 of=tfile cbs=16b
/* Mude a fita. */
% dd if=tfile of=/dev/st0 cbs=16b
% rm tfile
```

Obviamente, caso você tenha somente uma unidade de fita, precisará ter espaço em disco suficiente para armazenar uma imagem de toda a fita.

volcopy: sistemas de arquivos duplicados

volcopy é um utilitário do Linux que cria uma cópia exata de um sistema de arquivos em um outro dispositivo, modificando o tamanho de bloco conforme apropriado. Podemos usar **volcopy** para fazer o backup de um sistema de arquivos para um disco removível ou para criar uma cópia completa de um sistema de arquivos na fita. Consulte a página de manual em seu sistema para as opções e sintaxe apropriadas.

10.7 Usando Vários Arquivos em uma Única Fita

Na realidade, uma fita magnética contém uma longa string de dados. Porém, normalmente é útil armazenar mais do que uma "coisa" numa fita, de modo que as unidades de fita e seus drivers Linux conspirem para fornecer-lhe um pouco mais de estrutura. Quando **dump** ou outro comando grava um fluxo de bytes numa unidade de fita e depois fecha o arquivo de dispositivo, é automaticamente colocada uma marca de final de arquivo (EOF) na fita. Essa marca separa um fluxo de dados de outros fluxos de dados que são gravados subseqüentemente. Quando o fluxo é lido novamente, a leitura pára automaticamente em EOF.

Podemos usar o comando **mt** para posicionar uma fita num determinado fluxo ou "conjunto de arquivos", à medida que **mt** os chama. **mt** é especialmente útil se você colocar vários arquivos (por exemplo, vários dumps) numa única fita. Ele também possui algumas das mensagens de erro mais interessantes de qualquer utilitário Linux. O formato básico do comando é

```
mt [-f tapename] command [count]
```

Há várias opções para *command*; elas variam de plataforma para plataforma. Sendo assim, discutiremos somente aquelas que são essenciais para fazer backups e restaurações:

rew rebobina a fita para o início.

offl coloca a fita no estado off-line. Na maioria das unidades de fita, este comando faz com que a fita rebobine e saia da unidade. A maioria dos scripts usa este comando para ejetar a fita quando o processo tiver terminado, dando uma clara indicação de que tudo foi finalizado corretamente.

status imprime informações sobre o estado atual da unidade de fita (se a fita está carregada etc.).

fsf [*count*] avança rapidamente a fita. Se não for dada nenhuma contagem (*count*), **fsf** avança um arquivo. Com um argumento numérico, ele pula o número de arquivos especificados. Use este comando para avançar para o sistema de arquivos correto numa fita com vários dumps.

bsf [*count*] deve retornar o número de arquivos indicado em *count*. O comportamento exato desta diretiva depende do hardware da unidade de fita e seu driver associado. Em algumas situações, o arquivo atual é contado. Em outros, ele não é. Em alguns equipamentos, **bsf** não faz nada silenciosamente. Caso você avance muito em uma fita, sua melhor opção é rebobiná-la (**rew**) e começar desde o início.

Consulte seus manuais para obter uma lista exata dos comandos suportados por **mt**.

www.snow666.blogspot.com
Snow666

Robô a seu dispor?

Caso você seja afortunado o bastante para ter uma biblioteca de fitas robotizada, talvez você possa controlar seu trocador de fitas instalando o pacote **mtx**, uma versão melhorada do comando **mt**. Nós o usamos, por exemplo, para troca de fitas sem a intervenção de operadores em nosso maravilhoso sistema de cartuchos de fita HP 6x24 DAT. É só pra olhar, não para tocar!

10.8 AMANDA

O Amanda, Advanced Maryland Automatic Network Disk Archiver, é um sofisticado sistema de backup em rede capaz de substituir os scripts caseiros usados por muitas instalações. Ele é capaz de fazer o backup de todas as máquinas de uma rede numa única unidade de fita do servidor. O Amanda não somente é gratuito, como também roda numa série de sistemas UNIX e Linux. Ele é portanto uma excelente opção para fazer backups de instalações que usam uma mistura de sistemas operacionais.

O Amanda foi originalmente escrito por James da Silva da Universidade de Maryland em 1991. Agora ele é suportado por uma equipe de administradores de sistema ao redor do mundo. As informações mais recentes e o código-fonte se encontram disponíveis em www.amanda.org.

O Amanda não é um programa de backup em si, mas sim um envoltório que gerencia outros programas de backup. A maioria das instalações faz com que o Amanda use os comandos **dump** e **restore** nativos de seus sistemas como software de backup subjacente, porém o Amanda também pode orientar o **tar** e até mesmo **smbtar** do Samba para aquelas máquinas Windows irritantes.

O Amanda suporta uma ampla gama de unidades de fita e tira proveito de jukeboxes e empilhadores. O Amanda pode usar os recursos de compactação do hardware das unidades de fita ou compactar imagens de backup com **compress** ou **gzip** em máquinas-cliente antes de os dados trafegarem pela rede.

Um outro grande recurso do Amanda é o gerenciamento de fitas. O Amanda grava um cabeçalho em cada fita que ele usa, e portanto, jamais sobregravará a fita errada. Além disso, o Amanda mantém gerencia níveis de dump baseado em parâmetros de configuração e na integralidade das fitas. O Amanda mantém registros de quais backups estão em quais fitas e ele é capaz de imprimir etiquetas colantes mostrando o conteúdo de cada fita. (Essas etiquetas se tornam muito úteis quando o disco que armazena o banco de dados do Amanda cai.)

O Amanda é uma das soluções de backup gratuitas mais populares em uso comum. Ele atinge cerca de 1.500 instalações ao redor do mundo. Ele parece crescer bem e está sendo aprimorado cada vez mais para suportar os mais recentes e excelentes dispositivos para backup.

A arquitetura do Amanda

No modelo do Amanda, as unidades de fita e os discos que o armazenam são conectados a um servidor central. O servidor também abriga todos os arquivos de configuração, arquivos de log e bancos de dados do Amanda. O Amanda pode gravar somente uma imagem de backup em cada fita por vez, porém é capaz de fazer spool de vários dumps para seus discos simultaneamente e depois os colocar na fita.

O Amanda suporta várias "configurações" na mesma máquina servidora. Por exemplo, uma configuração poderia executar somente níveis 0 de clientes, ao passo que uma outra faria somente incrementais. Cada configuração gera os seus próprios arquivos de log e bancos de dados.

O servidor Amanda pesquisa seus arquivos de configuração para determinar quais sistemas de arquivos precisam ser colocados em backup, quais dispositivos de fita estão disponíveis e que quantidade de recursos de sistema (largura de banda da rede, fitas, carga de CPU etc.) é permitida para uso. A seguir ele contata máquinas-cliente e pergunta a elas para estimar o tamanho de seus arquivos de backup. Usando essas informações, o Amanda programa os backups.

O servidor Amanda é, na verdade, um conjunto de programas que implementa as diversas partes do sistema. É melhor executar os programas de servidor numa máquina rápida e que geralmente não esteja tão ocupada. Se estiver fazendo o backup de grandes quantidades de dados, o servidor deve ter a melhor conectividade que a arquitetura de seu sistema permitir. Já que o disco que abriga os arquivos do Amanda é usado para gerar spools de imagens de dumps da rede para a unidade de fita, ele deve ser pelo menos tão grande quanto a maior partição da qual pretende fazer um backup. O Amanda também precisa de algum espaço em disco (<75 MB) para seus próprios logs e bancos de dados.

Na época em que este livro estava sendo desenvolvido, a versão estável mais recente do Amanda era a 2.4.2p2; os exemplos, arquivos de configuração e comandos ilustrados neste capítulo foram extraídos dessa versão.

Snow666

Configuração do Amanda

Ao fazer o download do Amanda, use a versão estável mais recente, não aquela que se encontra atualmente em desenvolvimento. Estamos falando de backups, de modo que queremos algo realmente estável!

Após descompactar o código-fonte, leia os arquivos **README**, **docs/SYSTEM.NOTES** e **docs/INSTALL**. Quaisquer detalhes específicos a uma determinada arquitetura estão documentados no arquivo **SYSTEM.NOTES**. O arquivo **INSTALL** dá instruções passo a passo para instalação.

Antes de executar **configure**, execute **configure --help** para ver a lista completa de opções. Além de decidir onde o software deve ser instalado, você precisa definir qual usuário e grupo o Amanda executará. Para cada partição que você pretende fazer o backup com **dump**, o dispositivo bruto correspondente deve ser legível por este usuário. Os arquivos de dispositivo já estarão no grupo **disk** e, portanto, adicionar o usuário Amanda a este grupo fornecerá o acesso necessário.

Após executar **configure**, execute **make** e **make install** para completar a instalação.

Todo cliente Amanda precisa ter acesso aos binários do Amanda. Porém, não é uma boa idéia fazer com que todo cliente acesse os binários através de um sistema de arquivos baseado em rede, já que há oportunidades em que todos os clientes precisarão rodar simultaneamente (em particular no início do procedimento de backup, quando o Amanda solicita aos clientes que estimem seus tamanhos de dump). É melhor instalar os binários nos discos locais de cada cliente, normalmente em algum subdiretório de **/usr/local**.

Os programas a seguir devem ser instalados em todos os clientes. Eles jamais devem ser executados manualmente.

amandad manipula todas as comunicações entre o cliente e o servidor central; roda todos os demais programas-cliente.

selfcheck verifica se o cliente está configurado para o Amanda: se possui as permissões de dispositivo corretas, se é capaz de localizar **gzip**, se é capaz de gravar em **/etc/dumpdates** etc.

sendbackup realiza os backups.

sendsize estima os tamanhos de backup em diferentes níveis de dump.

As máquinas-cliente precisam de algumas outras configurações também. Cada dispositivo que desejamos fazer o backup deve ser legível pelo grupo do Amanda e o arquivo **/etc/dumpdates** tem de ser gravável por este grupo também. Assim que achar que tudo está configurado de acordo, use **amcheck** para verificar a configuração.

A seguir apresentamos uma lista dos comandos de servidor do Amanda. A maioria usa um argumento de linha de comando que os informa sobre que configuração Amanda usar.

amdump faz os dumps noturnos; normalmente executado por **cron**.

amflush despeja o disco retentor para a fita caso haja problemas.

amcleanup faz uma limpeza caso o hostmestre trave durante os dumps.

amrestore trata de todas as operações de restauração dos dumps do Amanda.

amlabel grava rótulos Amanda na fita; usado para evitar enganos como, por exemplo, gravar por cima de fitas erradas etc.

amadmin localiza a fita correta a ser restaurada e executa várias outras tarefas administrativas.

amcheck verifica se você está usando a fita correta (esperada), se há espaço livre suficiente no disco retentor e se os hosts-cliente estão configurados corretamente.

amtape gerencia empilhadores e trocadores de fitas.

amplot gera gráficos da atividade do Amanda (por exemplo, disco retentor e emprego da rede) para cada execução de dump.

Cada configuração é mantida num diretório distinto e precisa dos arquivos **amanda.conf** e **disklist**. O arquivo **amanda.conf** especifica a configuração geral do servidor, e o arquivo **disklist** especifica quais clientes e sistemas de arquivos devem ser colocados em backup. Esses arquivos residem somente no servidor.

O arquivo amanda.conf

amanda.conf é um arquivo bem grande, portanto o discutiremos em quatro blocos lógicos: informações locais, estratégia de dump, parâmetros de recursos e definições de tipos de dump. Esses blocos são pura invenção nossa; nem o Amanda nem a documentação do Amanda distingue as partes do arquivo **amanda.conf** dessa maneira.

O formato de arquivos é relativamente auto-explicativo, de maneira que o apresentamos na forma de um longo exemplo com comentários em cada seção. Começamos com a seção de informações locais que identifica a organização e o responsável pelo dump, define os arquivos de log e outros arquivos de configuração e especifica o formato dos rótulos das fitas (os rótulos lógicos gravados nas fitas em si, não as etiquetas colantes).

Snow666


```
#######################################################
# Parâmetros locais
#######################################################

org "Podunk Univ."      # o nome de sua organização para relatórios
mailto "amanda"         # lista de operadores de sua instalação separada por espaços
dumpuser "amanda"       # o usuário para executar dumps como
runtapes 1              # número de fitas a serem usadas num único ciclo de amdump
tpchanger "chg-manual" # script para dispositivo de mudança automática de fitas (fornecido pelo Amanda)
tapedev "/dev/nst0"     # o dispositivo de fita sem rebobinamento a ser usado

labelstr "^Podunk-[0-9][0-9]*$"  # regex para restrição de rótulo: todas as fitas têm que ser coincidentes

infofile "/usr/adm/amanda/podunk/curinfo" # diretório de bancos de dados
logdir "/usr/adm/amanda/podunk"           # diretório de logs
indexdir "/usr/adm/amanda/podunk/index"   # diretório de índices
```

O Amanda lê o rótulo de cada mídia e não usará uma fita a menos que o rótulo seja coincidente com a expressão regular especificada em `labelstr`. Portanto, todas as fitas têm de ser rotuladas por **amlabel** antes de poderem ser usadas para armazenar backups. Os rótulos não podem conter caracteres de espaço em branco.

Fica a critério do administrador de sistemas definir um esquema de nomes com significado. Poderíamos, por exemplo, ter * na concordância regex como uma dica para a compactação usada para o tipo de máquina do qual está sendo feito o backup. No exemplo acima, as fitas teriam rótulos Podunk-01, Podunk-02 e assim por diante.

Se tentarmos reutilizar uma fita antes de o ciclo de rodízio normal ter sido completado, o servidor a rejeitará. Esse recurso impede-nos de acidentalmente gravarmos por cima de uma fita que ainda é de importância histórica no esquema de backup definido.

Os parâmetros deste esquema (com que freqüência uma determinada fita pode ser usada, com que freqüência cada sistema de arquivos deve obter um dump nível 0, quando aumentar o nível do dump etc.) são definidos na segunda seção de **amanda.conf**.


```
#######################################################
# Parâmetros de estratégia
#######################################################

dumpcycle 4 weeks  # o número de dias no ciclo de dump normal
bumpdays 2         # número mínimo de dias em cada nível
bumpsize 20 Mb     # número mínimo de salvamentos (threshold) para mudar do nível 1 -> 2
bumpmult 2         # threshold = bumpsize * bumpmult^(level-1)
runspercycle 20    # o número de execuções de amdump em dias dumpcycle
                   # 4 semanas * 5 execuções de amdump por semana (dias úteis)
tapecycle 25 tapes # o número de fitas em rodízio
                   # 4 semanas (dumpcycle) * 5 fitas por semana (dias úteis)
                   # mais algumas para lidar com erros que precisem de amflush e
                   # para não sobregravarmos os backups completos realizados
                   # no início do ciclo anterior
```

No sistema Amanda, os dumps não são programados rigorosamente por data de calendário. Em vez disso, fornecemos ao Amanda informações genéricas sobre o nível de redundância desejado. O Amanda tenta distribuir o trabalho ao longo de todo o ciclo de dump para que as fitas sejam usadas de forma eficiente e a proteção seja maximizada. O Amanda recalcula seu cronograma diário a cada noite, usando informações em tempo real sobre cada sistema de arquivos cliente. Realmente não é possível predizer a agenda de dump exata com antecedência.

Um "ciclo de dump" é um período no qual os dumps nível 0 serão feitos pelo menos uma vez em cada sistema de arquivos. Os ciclos longos aumentam a flexibilidade de programação do Amanda, mas também podem aumentar o número médio de fitas que deve ser lido para completar uma restauração. O Amanda poderia realizar dumps nível 0 com mais freqüência do que o estritamente necessário caso haja fita disponível.

O modelo Amanda supõe que os dumps sejam feitos todo dia. Se isso não for verdade, você deve configurar o parâmetro `runspercycle` para o número real de dias de dump num ciclo. Esse exemplo de configuração realiza dumps em dias de semana, o que é razoável se alguém tem de estar fisicamente presente para trocar fitas.

Uma outra suposição que é feita por padrão é que será gravada uma fita de dump por dia (ou por "rodada", na terminologia do Amanda). Se estiver usando um empilhador ou jukebox, poderá gravar várias fitas por rodada. Será necessário uma configuração adicional.

www.snow666.blogspot.com
Snow666

O parâmetro `tapecycle` informa quantas fitas você planeja usar em rodízio regular. Um valor mínimo é o número de rodadas por ciclo de dump vezes o número de fitas por rodada, mais algumas extras para tratar de condições extremas e problemas com fitas. Se você alocar pelo menos o dobro do número mínimo de fitas, estará sempre garantido de ter em mãos pelo menos dois dumps nível 0 de cada sistema de arquivos.

Os parâmetros "bump" permitem que você exercite algum controle sobre o tamanho que pode atingir um dump incremental antes de o Amanda deslocar para o próximo nível de dump. Os parâmetros são especificados de uma maneira um tanto matemática e não intuitiva. Felizmente, podemos usar a opção **bumpsize** de **amadmin** para certificar-nos de que realmente implementamos a estratégia pretendida.

Por exemplo, com os parâmetros mostrados anteriormente e supondo-se que os arquivos de configuração sejam armazenados num diretório chamado **podunk**, **amadin** relataria o seguinte:

```
# amadmin podunk bumpsize

Parâmetros de bump atuais:
bumpsize 20480 KB - número mínimo de salvamentos (threshold) para mudar do nível 1 -> 2
bumpdays 2        - número mínimo de dias em cada nível
bumpmult 2        - threshold = bumpsize * (level-1)**bumpmult

Bump -> To    Threshold
1    -> 2      20480 KB
2    -> 3      40960 KB
3    -> 4      81920 KB
4    -> 5     163840 KB
5    -> 6     327680 KB
6    -> 7     655360 KB
7    -> 8    1310720 KB
8    -> 9    2621440 KB
```

Após um nível 0 inicial, o Amanda iniciará fazendo dumps nível 1. Assim que os dumps nível 1 se tornarem maiores do que 20 MB, o Amanda mudará para o nível 2. Assim que os dumps nível 2 se tornarem maiores do que 40 MB, o Amanda transferirá para o nível 3 e assim por diante.

Você deve ajustar cuidadosamente esses parâmetros para obter um balanceamento entre o nível de redundância e o custo de fitas. Redundância excessiva resultará em custos operacionais maiores e redundância muito pequena poderá resultar em perdas de dados.

O restante do arquivo **amanda.conf** é dedicado a parâmetros que especificam quanta largura de banda, CPU e espaço em disco (no servidor) devem ser usados; o tipo de unidade de fita para backup e os tipos de partições de cliente a serem colocados em backup.

```
#######################################################
# Parâmetros de recursos
#######################################################

tapetype EXB-8500 # que tipo de fita ela é (refira-se aos tipos de fita abaixo)
inparallel 4      # número máximo de dumpers de clientes que serão executados em paralelo
netusage 600 Kbps # largura de banda de rede máxima para o Amanda, em KB por segundo
etimeout 300      # número de segundos a aguardar para estimativa por sistema de arquivos

holdingdisk hd1 {
     comment "disco retentor principal"
     directory "/dumps/amanda"              # ponto de montagem do disco retentor
     use 8196 Mb                            # quanto espaço podemos usar nele
}

define tapetype EXB-8500 {
     comment "Drive Exabyte EXB-8500 em máquina apropriada"
     length 4200 mbytes
     filemark 48 kbytes
     speed 474 kbytes
}
```

Snow666

Esse exemplo ilustra a configuração para uma unidade de fita Exabyte 8500. O tipo de parâmetros de fita é extremamente importante e jamais devem ser supostos. Se sua unidade de fita não estiver listada no arquivo **amanda.conf** modelo que vem junto com a distribuição, provavelmente poderá encontrá-la no arquivo **docs/TAPETYPES** ou em

http://www.cs.columbia.edu/~sdossick/amanda

Talvez você também deva tentar solicitar uma configuração apropriada em uma das listas de discussão do Amanda. Se tudo isso não funcionar, você poderá usar o programa **tapetype** que vem junto com o Amanda. Ele determinará os parâmetros corretos para sua unidade de fita preenchendo uma fita com blocos de 32 KB. Entretanto, apesar desse procedimento ser um recurso de último caso — ele pode levar um bom tempo (de 1 a 2 dias!) em algumas unidades de fita.

Os parâmetros de recurso finais são os tipos de dumps, que representam os diversos tipos de dados (por exemplo, voláteis, importantes, estáticos etc.) que um sistema de arquivos pode conter. Cada sistema de arquivos num cliente tem de ter atribuído a si um tipo de dump em particular. O tipo de dump também especifica que tipo de compactação (se realmente existir alguma) deve ser aplicado quando os dados são arquivados. Eis alguns exemplos:

```
##########################################################
# Definições de tipos de dumps
##########################################################

define dumptype comp-user {
     comment "partições em máquinas razoavelmente rápidas"
     compress client fast
     priority medium
}

define dumptype comp-root {
     comment "partições-raiz em máquinas razoavelmente rápidas"
     compress client fast
     priority low
}

define dumptype nocomp-user {
     comment "partições em máquinas lentas"
     compress none
     priority medium
}

define dumptype clone-user {
     comment "partições que não devem receber incrementos"
     compress client fast
     skip-incr
     priority medium
}

define dumptype comp-high-samba {
     comment "usado para sistemas de arquivos Windows"
     program "GNUTAR"
     compress server fast
}

define dumptype dos-user {
     comment "usado para partições DOS que são montadas sempre"
     program "GNUTAR"
     compress client fast
}
```

Esses tipos de dump em particular são todos predefinidos pelo Amanda. Podemos usá-los diretamente, personalizá-los para nossos próprios objetivos ou escrevermos nossas próprias especificações do zero. O campo de comentário ajuda a indicar o que cada tipo de dump faz.

www.snow666.blogspot.com
Snow666

O parâmetro `compress` especifica onde os dados de dump são compactados: no cliente, no servidor ou em algum outro ponto. Especificamos o programa de compactação (por exemplo, **compress** ou **gzip**) ao instalarmos inicialmente o Amanda. Os valores possíveis para o Amanda são `none`, `client best`, `client fast`, `server best` e `server fast`. O padrão é `client fast`.

`compress none` desativa a compactação. Poderíamos escolher essa opção caso a unidade de fita do servidor implementasse compactação via hardware. As opções `client` compactam os dumps antes de eles serem enviados através da rede ao servidor e as opções `server` compactam os dumps assim que eles atingem o disco retentor do servidor. Se quiser usar compactação de software normalmente é indicado executar a compactação nos clientes.

Os modificadores `best` e `fast` informam o compactador com que nível efetuar a compactação de dados; compare **`gzip --best`** e **`gzip --fast`**. Só usamos `fast`; `best` pode demorar muito mais e a compactação não é tão melhor assim.

O parâmetro `holdingdisk` possui dois valores possíveis: `yes` e `no`. Ele especifica se o disco retentor deve ser usado para gerar spool desse backup. Devemos desativar essa opção ao arquivarmos o disco retentor em si. O valor-padrão é `yes`.

O parâmetro `maxdumps` especifica o número máximo de dumps concorrentes que podem ser executados num cliente. O valor padrão é 1, porém poderíamos aumentá-lo para melhor desempenho em servidores de arquivos robustos que possuem muito recurso de CPU e largura de banda.

O parâmetro `priority` informa a importância do backup. Os valores possíveis são `low`, `medium` e `high`. O padrão é `medium`. Se não houver fita suficiente para armazenar todos os dumps que foram programados, os dumps de menor prioridade serão pulados. No caso de um erro na fita, o Amanda tenta direcionar os backups de maior prioridade para o disco retentor; caso haja espaço, os backups de prioridade menor também serão colocados lá.

Recomendamos definir um tipo de dump distinto para cada prioridade. Os diretórios de usuário devem ter um dump de alta prioridade. A prioridade média é indicada para pacotes de software local (por exemplo, **/usr/local**) e a prioridade baixa para arquivos de sistema que não mudam com tanta freqüência.

O parâmetro `program` especifica se devemos usar **dump** ou **tar**. O padrão é **dump** e esta normalmente é a melhor opção.

A opção `skip-full` instrui o Amanda para pular o sistema de arquivos quando um dump nível 0 for recomendado. Geralmente selecionaríamos essa opção quando dumps nível 0 forem executados fora do Amanda. Poderíamos, por exemplo, optar por fazer dumps nível 0 somente quando a máquina estiver no modo monousuário.

A opção `skip-incr` faz com que o Amanda pule todos os dumps exceto os de nível 0. Devemos escolher essa opção para configurações de arquivamento nas quais somente dumps completos são feitos e as fitas são salvas por tempo indeterminado.

O arquivo disklist

O arquivo **amanda.conf** informa como fazer dumps sem na verdade especificar nenhum cliente ou sistema de arquivos para dump. Essas informações são registradas no arquivo **disklist**. É atribuído um dos tipos de dump definidos em **amanda.conf** a cada sistema de arquivos cliente.

```
#####################################################################
# cliente    partição     dumptype           # ponto de montagem
#####################################################################

# o servidor de dumps
ocean        sd0a         comp-root       # /
ocean        sd0g         comp-user       # /usr
ocean        sd0d         comp-user       # /var
ocean        sd0h         comp-high       # /amanda

# Partição Windows de lorien montada via Samba em ocean
ocean     //lorien/c$  comp-high-samba    # c:\

# protótipo
squish       yc0t0d0s0    comp-high       # /
squish       yc0t0d0s6    comp-high       # /usr
squish       yc0t0d0s3    comp-high       # /var
squish       yc0t0d0s7    comp-high       # /local

# clone
zamboni      hda1         clone-user      # /
```

Snow666


```
zamboni     hda6        clone-user      # /usr
zamboni     hda5        comp-root       # /var
zamboni     hda3        comp-user       # /local

# PC lento
fuzz     sdla        nocomp-high       # /
fuzz     sdlf        nocomp-high       # /local
fuzz     sdle        nocomp-high       # /usr
fuzz     sdld        nocomp-high       # /var
fuzz     /dos        dos-user          # /dos
```

A primeira coluna é o nome de host da máquina-cliente da qual deve ser feito o backup. A segunda coluna lista a partição de disco desejada. Podemos usar o nome de dispositivo (como feito no exemplo) ou então o ponto de montagem.

Note que o disco retentor do servidor de dump (ocean), **/dumps/amanda**, não é mencionado no arquivo **disklist**. Em nosso caso não é necessário fazer backup dessa partição, pois o seu conteúdo é limitado a imagens de dump armazenadas durante execuções do Amanda. Se mantivermos logs Amanda ou outras informações importantes no disco retentor, devemos fazer o seu backup usando o parâmetro de tipo de dump `holdingdisk`.

A opção `skip-incr` (incluída no tipo `clone-user` definido no exemplo) é prática para fazer o backup de clones de uma máquina-protótipo. Em nossa instalação temos uma máquina-protótipo para cada arquitetura que é clonada em outras máquinas da mesma arquitetura. Já que as partições-raiz dos clones são as mesmas que as de seus protótipos, não precisamos desperdiçar espaço em disco fazendo backup delas toda noite.

Porém, cada clone possui alguns arquivos exclusivos (por exemplo, arquivos de configuração em **/etc**), de forma que ainda fazemos um nível 0 durante o ciclo de dumps. Se você tiver uma configuração similar e optar por não fazer backups todas as noites de determinados sistemas de arquivos, certifique-se de que eles sejam de fato idênticos a algo que é arquivado todas as noites. Em nosso exemplo, a partição **/var** de zamboni armazena o e-mail dos usuários, de modo que ela tenha de ser colocada em backup todas as noites.

Consulte o Capítulo 27 para obter maiores informações sobre o Samba.

Instalamos o programa **smbtar** do Samba em nosso servidor de dumps para podermos fazer backup de sistemas de arquivos do Windows. Nesse exemplo de configuração, fazemos backup do drive C: de lorien. Note que o host-cliente listado no arquivo **disklist** é ocean, não lorien. Se estiver usando o Samba para acessar um sistema de arquivos, o cliente Amanda terá de ser o host Linux no qual **smbtar** está localizado, e não a máquina Windows. (Não especificamos a partição **/dos** de fuzz dessa maneira, pois ela sempre é montada e não é acessada através do Samba.) O Amanda faz a distinção entre partições Samba e pontos de montagem comuns (por exemplo, **/usr** ou **/dos**) pelo número de barras no início: duas para o Samba e uma para pontos de montagem comuns.

Arquivos de log do Amanda

O Amanda cria dois arquivos no servidor para cada execução. O primeiro é **amdump**.*n*, onde *n* é o número de execuções Amanda adicionais que ocorreram desde que o arquivo de log foi criado. O arquivo contém uma descrição verbal das decisões que o Amanda tomou. O outro arquivo de log é **log**.*date*.*n*, onde *date* é a data da execução e *n* é o número de execuções já realizado naquele dia.

Depuração do Amanda

Após cada execução, o Amanda gera um relatório sintetizado das atividades da execução e envia um e-mail ao responsável pelos dumps. Esse relatório inclui informações sobre a quantidade de fita usada, os sistemas de arquivos cujos backups foram realizados com sucesso e quaisquer erros que forem encontrados. Eis um exemplo de uma configuração usando uma versão ligeiramente diferente de **disklist** do que aquela mostrada anteriormente:

```
To: amanda@ocean
Subject: Podunk Univ. AMANDA MAIL REPORT FOR September 5, 2001

These dumps were to tape Podunk-481.
Tonight's dumps should go onto 1 tape: Podunk-482.

FAILURE AND STRANGE DUMP SUMMARY:
   fuzz sdla lev 0 FAILED [no estimate or historical data]
...
   taper: FATAL syncpipe_get: w: unexpected EOF
```

Snow666


```
STATISTICS:                Total     Full      Daily
                           ------    ------    ------
Dump Time (hrs:min)        3:02      0:36      0:04 (0:34 start, 1:49 idle)
Output Size (meg)          2954.6    2666.8    287.8
Original Size (meg)        7428.1    6292.5    1135.5
Avg Compressed Size (%)    39.8      42.4      25.3
Tape Used (%)              70.5      63.5      7.0 (level:#disks …)
Filesystems Dumped         18        8         10  (1:8 2:2)
Avg Dump Rate (k/s)        105.3     124.5     43.4
Avg Tp Write Rate (k/s)    1254.2    1251.8    1276.9
NOTES:
  planner: Adding new disk zamboni:hda3.
  driver: WARNING: /dumps/amanda: 8550400 KB requested, but only 1035113
    KB available.
  planner: Forcing full dump of squishy:c0t0d0s0 as directed.
  planner: Request to fuzz timed out.
  planner: Incremental of ocean:sd0h bumped to level 2.
…
  driver: going into degraded mode because of tape error.
…
```

Um problema comum é que o Amanda talvez não seja capaz de gravar backups numa fita. Esse problema pode ocorrer quando não há uma fita válida na unidade ou quando algum tipo de erro de fita ocorre durante a gravação (como no exemplo com o host fuzz). Em ambos os casos, o Amanda ainda envia spools do backup para o disco retentor. Para gravar esses dumps jogados em buffer de forma definitiva para fita, insira a fita apropriada na unidade e execute **amflush**.

Para diagnosticar outros problemas, você poderá investigar os arquivos de log no servidor ou depurar arquivos no cliente. A localização de arquivos de log no servidor é especificada no arquivo **amanda.conf**. Os arquivos de depuração estão localizados no diretório **/tmp/amanda** de cada cliente, isto é, caso compilemos Amanda com **--with-debugging** (o padrão).

O Amanda gera o relatório via e-mail por meio do arquivo de log. Eis, por exemplo, o arquivo de log (**amdump**.*n*) com base no qual o e-mail anterior foi derivado.

```
SETTING UP FOR ESTIMATES…
dumper: pid 18199 executable dumper version 2.4.1p1, using port 791
…
driver: started dumpersetup_estimates: ocean:sd0d: command 0, options:
  last_level 1 next_level0 6 level_days 16
  getting estimates 0 (20023) 1 (2735) -1 (-1)
…
zamboni:hda1 lev 1 skipped due to skip-incr flag
planner: SKIPPED zamboni hda1 [skip-incr]
…
GETTING ESTIMATES…
got result for host ocean disk sd0a: 0 -> 53797K, 1 -> 1797K, -1 -> -1K
got result for host ocean disk sd0d: 0 -> 19695K, 1 -> 2696K, -1 -> -1K
…
ANALYZING ESTIMATES…
…
pondering ocean:sd0d… next_level0 6 last_level 1 (not due for a full dump,
   picking an incr level)
…
```

Eis o outro arquivo de log, **log.20010905.0**, que corresponde ao e-mail original:

```
START planner date 20010905
START driver date 20010905
INFO planner Adding new disk depot:dsk/d1.
SUCCESS planner zamboni hda1 1 [skipped: skip-incr]

WARNING driver WARNING: /dumps/amanda: 8550400 KB requested, but only
   1035113 KB available.
```

Snow666


```
...
START taper datestamp 20010905 label Podunk-481 tape 0
FAIL planner fuzz sd1a 0 [no estimate or historical data]

...
STATS driver startup time 2019.456
SUCCESS dumper ocean sd0a 0 [sec 418.311 kb 25088 kps 59.97 orig-kb 58087 ]
SUCCESS dumper ocean sd0d 1 [sec 15.867 kb 800 kps 50.42 orig-kb 2719 ]

...
SUCCESS taper ocean sd0a 0 [sec 53.366 kb 25088 kps 474.612 {wr: writes 2
   rdwait 0.000 wrwait 0.032 filemark 38.332 }]
SUCCESS taper ocean sd0d 1 [sec 6.345 kb 800 kps 133.3 {wr: writes 1 rdwait
   1.470 wrwait 0.356 filemark 2.637 }]

...
STRANGE dumper ocean sd0h 1 [sec 82.435 kb 33.4 kps 0.4 orig-kb 155.0 ]
  sendbackup: start [ocean:sd0h level 1 datestamp 20010905]
  | DUMP: Date of this level 1 dump: Wed Sep 05 23:47:54 2001
  | DUMP: Date of last level 0 dump: Mon Aug 27 23:43:23 2001
  | DUMP: Dumping /dev/rsd0h (/amanda) to standard output
  | DUMP: mapping (Pass I) [regular files]
  | DUMP: mapping (Pass II) [directories]
  ? DUMP: (This should not happen) bread from /dev/rsd0h [block 64]:
    count=8192, got=-1
  | DUMP: estimated 38 blocks (19KB) on 0.00 tape(s).
  | DUMP: dumping (Pass III) [directories]
  | DUMP: dumping (Pass IV) [regular files]
  | DUMP: level 1 dump on Wed Sep 05 23:47:54 2001
  | DUMP: 310 blocks (155KB) on 1 volume
  | DUMP: DUMP IS DONE
  sendbackup: size 158720
  sendbackup: end
...
```

Cada linha `SUCCESS dumper` significa que um dump foi gravado no disco retentor, e cada linha `SUCCESS taper` significa que um dump foi gravado na fita. Um `STRANGE dumper` significa que o Amanda observou saída de erro enquanto executava **dump**. Quando ocorrem erros de dump, o Amanda salva a saída nesse arquivo de log (ele coloca um "?" na frente das linhas problemáticas) e também o inclui no sumário de e-mail.

Um outro problema comum ocorre quando o Amanda não é capaz de obter uma estimativa dos tamanhos dos dumps para os sistemas de arquivos de um cliente. As primeiras coisas a serem verificadas nessa situação são se o cliente pode ser acessado via rede e se a cópia do Amanda desse cliente está configurada apropriadamente. Também podemos verificar os arquivos de depuração no diretório **/tmp/amanda** do cliente. Cada um dos programas-cliente do Amanda grava saída de depuração nesse local toda vez em que ele é executado.

Quando o Amanda não consegue obter uma estimativa de tamanho, verifique o arquivo de depuração de **sendsize**. **sendsize** analisa sintaticamente a saída gerada por **dump** e procura a linha de tamanho estimado. Se ele não a encontrar, o Amanda relata `[no estimate]`. Eis um exemplo de um arquivo **sendsize.debug**.

```
sendsize: getting size via dump for sd5c level 1
sendsize: running "/usr/ccs/bin/dump 1sf 100000 - /dev/sd5c"
  DUMP: Date of this level 1 dump: Wed Sep 05 21:59:36 2001
  DUMP: Date of last level 0 dump: Mon Aug 27 05:08:33 2001
  DUMP: Dumping /dev/sd5c (/var) to standard output
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: mapping (Pass II) [directories]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 7150 tape blocks on 0.00 tape(s).
  DUMP: dumping (Pass III) [directories]
...

calculating for amname 'sd3d', dirname '/local'
```

Snow666


```
sendsize: getting size via dump for sd3d level 0
sendsize: running "/usr/ccs/bin/dump 0sf 100000 - /dev/sd3d"
  DUMP: Cannot open/stat /dev/sd3d, Permission denied
...
(no size line match in above dump output)
```

Neste caso, é óbvio que precisamos corrigir as permissões em **/dev/sd3d**.

Caso não consiga solucionar o problema analisando os arquivos de log ou a documentação do Amanda, talvez seja interessante analisar os arquivos das listas de discussão do Amanda. Você pode encontrá-los em

http://www.egroups.com/list/amanda-users/http://www.egroups.com/list/amanda-hackers

Restauração de arquivos de um backup Amanda

Para restaurar por meio de backups Amanda, use os programas **amadmin** e **amrestore**. Analisemos um exemplo completo. Suponha que tenhamos um usuário que eliminou um diretório inteiro e o queira restaurado. A primeira etapa é localizar as fitas no qual o diretório foi colocado em backup. Para tanto, você precisará das seguintes informações:

- O nome da máquina e a partição na qual o diretório residia.
- O caminho completo para este diretório.
- A data em que o diretório foi perdido ou corrompido.
- A data em que o diretório foi modificado pela última vez.

As datas determinarão o intervalo de fitas que contêm o diretório num estado que possivelmente vamos querer restaurar. Suponhamos que o diretório que queremos restaurar se encontre na partição **/local** de zamboni e que ele foi modificado em 5 de outubro e eliminado em 12 de outubro. **amadmin** determinará que fitas precisaremos:

```
% amadmin podunk find zamboni hda3
date        host     disk  lv tape        file  status
2002-01-26  zamboni  hda3  1  Podunk-795  33    OK
2002-01-25  zamboni  hda3  1  Podunk-794  41    OK
2002-01-23  zamboni  hda3  0  Podunk-792  9     OK
2002-01-22  zamboni  hda3  1  Podunk-791  32    OK
...
2001-10-13  zamboni  hda3  1  Podunk-685  38    OK
2001-10-12  zamboni  hda3  1  Podunk-684  37    OK
2001-10-11  zamboni  hda3  1  Podunk-683  39    OK
2001-10-10  zamboni  hda3  1  Podunk-682  72    OK
2001-10-09  zamboni  hda3  1  Podunk-681  44    OK
2001-10-08  zamboni  hda3  1  Podunk-680  88    OK
2001-10-07  zamboni  hda3  1  Podunk-518  35    OK
2001-10-06  zamboni  hda3  1  Podunk-517  33    OK
2001-10-05  zamboni  hda3  1  Podunk-516  33    OK
2001-10-04  zamboni  hda3  1  Podunk-515  51    OK
2001-10-03  zamboni  hda3  1  Podunk-514  16    OK
2001-10-02  zamboni  hda3  1  Podunk-513  19    OK
2001-10-01  zamboni  hda3  1  Podunk-512  36    OK
2001-09-30  zamboni  hda3  1  Podunk-511  15    OK
2001-09-29  zamboni  hda3  1  Podunk-510  78    OK
2001-09-28  zamboni  hda3  0  Podunk-509  99    OK
...
```

A opção **find** faz com que **amadmin** percorra cada arquivo de log na configuração **amanda.conf**, procurando pela máquina e partição que fornecemos como argumentos. Para efetuarmos nossa restauração, precisamos da fita nível 0 Podunk-509 e da fita nível 1 Podunk-683 do dia anterior à eliminação do diretório. Para o nível 1 poderíamos, teoricamente, usar qualquer fita gravada após o dia 5, já que o usuário diz que o diretório não foi modificado entre os dias 5 e 12. Porém, na ausência de uma razão para agir de outra maneira é melhor usar a fita mais recente disponível. Os usuários nem sempre têm boa memória.

A seguir, usamos **amrestore** para efetivamente restaurar os dados. Devemos começar com a fita Podunk-509 já que ela é de nível 0. Após inserirmos a fita na unidade de fita, poderíamos executar o seguinte

Snow666


```
% amrestore -p /dev/st0 zamboni hda3 | restore -i -f -
```

para localizar o registro apropriado e iniciar uma restauração interativa. **amrestore** procura cada imagem de dump na fita até localizar aquela que estamos procurando, depois envia essa imagem de dump para a sua saída-padrão para ser lida pelo comando **restore**. O procedimento de restauração em si funciona exatamente como descrito na seção 10.4 . Após extrair a fita nível 0, devemos repetir o processo com a fita de nível 1.

amrestore é capaz de reconhecer o dump apropriado, pois o Amanda coloca um cabeçalho de 32 K em frente a cada dump que registra de onde ele proveio e como ele foi compactado. Se foi usada compactação, **amrestore** direciona automaticamente a imagem da fita através do descompactador apropriado.

Localizar todos os cabeçalhos pode levar um bom tempo, já que **amrestore** teria de pular centenas de sistemas de arquivos. A saída de **amadmin** mostra qual conjunto de arquivos estamos procurando, para que realmente não seja necessário deixar **amrestore** ter todo este trabalho. Podemos simplesmente executar **mt fsf** para avançar a fita rapidamente antes de executarmos **amrestore**.

Temos de executar o processo de restauração (**restore**) no mesmo sistema operacional e arquitetura que geraram o dump original. O Amanda não sabe nada em relação ao conteúdo em si do dump e não é capaz de isolá-lo de problemas de compatibilidade entre plataformas diferentes.

Se os logs do Amanda forem eliminados, sempre podemos procurar nos rótulos das fitas para localizar as fitas de que precisamos. Porém o que aconteceria se **amrestore** em si fosse eliminado? Não se preocupe: **amrestore** não faz nada mais além do que **dd**. Se observarmos o cabeçalho de 32 K do backup que queremos restaurar, veremos instruções para restaurá-lo. Eis, por exemplo, o cabeçalho da fita nível 0:

```
# mt -f /dev/nst0 fsf 99
# dd if=/dev/nst0 bs=32k count=1
AMANDA: FILE 20010928 zamboni hda3 lev 0 comp .gz
To restore, position tape at start of file and run:
dd if=<tape> bs=32k skip=1 | gzcat | restore …f -

1+0 records in
1+0 records out
```


Alternativas para o Amanda: outros pacotes de backup com fonte aberto

Várias outras ferramentas de backup shareware ou gratuitas se encontram disponíveis na Internet para download. Algumas que valem a pena serem citadas são:

- Arkeia — uma ferramenta muito interessante baseada em interface gráfica com o usuário com excelente suporte
- BURT — um utilitário para backup e recuperação baseado em Tcl/TK 8.0
- **cddump** — um utilitário automatizado desenvolvido especificamente para CD-Rs
- **star** — uma implementação mais veloz de **tar**. Possui alguns recursos adicionais como swapping de bytes automático. Não afetará arquivos existentes na restauração.

10.9 Produtos Comerciais para Backup

Todos nós gostaríamos de imaginar o Linux como o único sistema operacional existente no mundo, porém, infelizmente, este não é o caso. Ao pesquisar soluções de backup comerciais, você deve considerar a capacidade deles em manipular quaisquer outros sistemas operacionais pelos quais você é responsável por fazer backups. A maior parte dos produtos contemporâneos leva em consideração a questão de plataformas diferentes e permite que incluamos estações de trabalho UNIX, Windows e Macintosh em nosso esquema de backup Linux. Também temos de considerar arrays de armazenamento não Linux e servidores de arquivos.

Os laptops dos usuários e outras máquinas que não estão conectadas consistentemente à rede também deveriam ser protegidos contra falhas. Ao pesquisar produtos comerciais, talvez você queira perguntar se cada produto é inteligente o bastante para não fazer backups de arquivos idênticos de cada laptop. De quantas cópias de **command.com** você realmente precisa?

Há vários pacotes comerciais comuns que talvez você queira investigar ao selecionar um sistema de backup abrangendo toda a instalação. Já que acabamos achando o Amanda adequado para o nosso caso, não temos muita experiência com produtos comerciais. Solicitamos as impressões de alguns de nossos companheiros mais afortunados em termos de recursos financeiros em instalações comerciais sobre os sistemas que eles adotam. Seus comentários são reproduzidos aqui.

Snow666

ADSM/TSM

O produto ADSM foi desenvolvido pela IBM e posteriormente adquirido pela Tivoli. Atualmente ele é comercializado como TSM (Tivoli Storage Manager). O TSM é uma ferramenta para gerenciamento de dados que também trata de backups. Maiores informações sobre o produto poderão ser encontradas em www.tivoli.com[11].

Prós:

- Apoiado pela IBM; veio ao mercado para ficar
- Preço atraente e opções de leasing
- Baixa taxa de falhas
- Usa cache em disco; útil para backups de clientes lentos
- Lida com clientes Windows
- Excelente documentação (cobrada à parte)

Contras:

- Interface gráfica com o usuário fraca
- Cada 2 arquivos equivalem a 1 K no banco de dados
- O projeto é incremental para sempre

Veritas

A Veritas comercializa soluções para backup para uma série de sistemas diferentes. Ao visitar o site Web da Veritas (www.veritas.com), certifique-se de selecionar o produto que é adequado para você.

Prós:

- Interface gráfica com o usuário decente
- Conecta-se diretamente a dispositivos da Network Appliance
- Instalação dinâmica para Linux
- Capaz de gravar fitas em formato **tar** GNU
- Banco de dados centralizado, porém capaz de suportar um sistema de backups distribuído

Contras:

- Alguns bugs
- O sistema de preços é confuso e irritante
- O suporte a Windows é pontual

Legato

O software para backup da Legato é vendido diretamente pela Legato, mas também é incorporado por alguns fabricantes de computadores importantes como a Compaq. Maiores informações poderão ser obtidas em www.legato.com.

Prós:

- Excelente interface gráfica com o usuário
- Preço bastante razoável
- E-mail automático para usuários informando-os sobre o estado dos backups

Contras:

- Ouvimos falar de problemas com um arquivo de índices corrompido
- Não recomendado para mais de 100 clientes
- Talvez não suporte bem clientes heterogêneos (embora seja veiculado que suporta)
- Ouvimos falar de problemas com sistemas de arquivos grandes
- Suporte sofrível

---

11. O site www.tivoli.com é redirecionado para uma área dentro do site da própria IBM. O produto TSM atualmente é comercializado pela IBM como IBM TSM (N.R.T.).

Snow666

Outras alternativas

W. Curtis Preston, autor do livro sobre backups da O'Reilly, mantém uma página Web muito útil sobre tópicos relacionados com backup (produtos de espelhamento de discos, produtos sobre sistemas de arquivos avançados, produtos sobre backups de sistemas remotos, produtos para passagem de dados fora da instalação etc.). Entre outros recursos, ele inclui uma tabela abrangente sobre cada software comercial para backup conhecido. Recomendamos enfaticamente tanto o livro como o site Web. O endereço é www.backupcentral.com.

10.10 Leitura Recomendada

PRESTON, Curtis W. *Unix Backup and Recovery*. O'Reilly, 1999.

10.11 Exercícios

E10.1 Investigue o procedimento para backups usado em sua instalação. Que máquina(s) executa(m) os backups? Que tipos de dispositivos de armazenamento são usados? Onde as fitas são armazenadas? Sugira melhorias ao sistema atual.

E10.2 Que etapas são necessárias para restaurar arquivos num sistema que usa o Amanda? Como localizar a fita correta?

★ **E10.3** Elabore um plano para backups para as seguintes situações. Suponha que cada computador tenha um disco de 10 GB e que os diretórios dos usuários são armazenados localmente. Selecione um dispositivo para backup que apresente um equilíbrio entre custo *versus* necessidades de suporte e explique sua motivação. Liste quaisquer suposições feitas.

a) Uma instalação de pesquisa com 50 máquinas. Cada máquina armazena uma grande quantidade de dados importantes que se altera com freqüência.

b) Uma pequena empresa de software com 10 máquinas. Os códigos-fonte são armazenados num servidor central com 50 GB de espaço em disco. Os fontes são alterados constantemente ao longo do dia. Os diretórios dos usuários não mudam muito. O custo não é uma preocupação e a segurança é de fundamental importância.

c) Uma rede doméstica com duas máquinas. O custo é uma questão importante.

★ **E10.4** Elabore uma estratégia de restauração para cada uma das três situações descritas acima.

★ **E10.5** Descreva as etapas que você usaria para realizar um **rdump** seguro através de um túnel SSH seguro.

★ **E10.6** Dada a saída a seguir gerada por **df** e **/etc/dumpdates**, identifique as etapas necessárias para realizar as três restaurações solicitadas. Suponha que a data de restauração solicitada seja 18 de janeiro.

A saída gerada por **df** da máquina khaya.cs.colorado.edu:

```
/dev/hda8      256194     81103    161863    33%   /
/dev/hda1       21929      4918     15879    24%   /boot
/dev/hda6     3571696     24336   3365924     1%   /local
/dev/hda10     131734      5797    119135     5%   /tmp
/dev/hda5     1815580   1113348    610004    65%   /usr
/dev/hda7      256194     17013    225953     7%   /var
```

/etc/dumpdates de khaya.cs.colorado.edu:

```
/dev/hda8      2 Thu Jan  17   22:59:23 2002
/dev/hda6      3 Thu Jan  17   22:51:51 2002
/dev/hda7      3 Thu Jan  17   22:50:24 2002
/dev/hda5      9 Thu Jan  17   22:46:25 2002
/dev/hda5      1 Sat Jan  17   22:45:42 2002
/dev/hda7      0 Sat Jan  12   23:14:47 2002
/dev/hda6      1 Sat Jan  12   23:14:32 2002
/dev/hda8      1 Sat Jan  12   23:14:17 2002
/dev/hda6      0 Thu Jan  10   22:47:31 2002
/dev/hda1      1 Tue Jan   8   22:16:05 2002
/dev/hda7      1 Mon Jan   7   22:08:09 2002
/dev/hda1      4 Thu Jan   3   22:51:53 2002
/dev/hda7      2 Mon Dec  24   22:53:52 2001
/dev/hda5      0 Sat Nov   3   22:46:21 2001
/dev/hda1      0 Fri Sep  21   22:46:29 2001
/dev/hda8      0 Fri Aug  24   23:01:24 2001
/dev/hda1      3 Sun Jul  29   22:52:20 2001
/dev/hda6      2 Sun Jul  29   23:01:32 2001
```

Snow666

a) "Favor restaurar meu diretório de usuário inteiro (**/usr/home/clements**) referente a algum momento nos últimos dias. Parece que perdi toda a base de código do meu principal projeto."
b) "Uumm, acidentalmente dei um **sudo rm -rf /*** na minha máquina khaya. Você poderia por favor restaurar todos os sistemas de arquivos a partir dos backups mais recentes?"
c) "Todos os meus arquivos MP3 que vinha colecionando de Aimster ao longo do último mês foi para o espaço. Eles estavam armazenados em **/tmp/mp3**. Você poderia restaurá-los para mim?"

★★ **E10.7** Escreva instruções de configuração para o Amanda que implementem planos de backup que você planejou para o exercício 10.3. Certifique-se de especificar os dispositivos de backup, os tipos de partição e os arquivos **disklist**.

Snow666

11 Syslog e Arquivos de Log

Os daemons de sistema, o kernel e os diversos utilitários todos emitem dados que são registrados e eventualmente acabam em nossos discos de tamanho finito. A maioria desses dados possui vida útil limitada e precisa ser sintetizada, compactada, arquivada e eventualmente jogada fora.

11.1 Políticas de Registro

As políticas de registro (logging) variam de instalação para instalação. Esquemas comuns incluem:

- Jogar fora todos os dados imediatamente.
- Reinicializar arquivos de log em intervalos periódicos.
- Fazer um rodízio dos arquivos de logs, mantendo dados por um período fixo.
- Compactar e arquivar logs em fita ou outras mídias permanentes.

A escolha correta para a sua instalação dependerá de quanto espaço em disco você tem e quão consciente você é em termos de segurança. Até mesmo instalações com uma abundância de espaço em disco têm de lidar com o crescimento cancerígeno dos arquivos de log.

Seja qual for o esquema adotado, você deve automatizar a manutenção de arquivos de log através de cron. Veja o Capítulo 9, *Processos Periódicos*, para obter maiores informações sobre esse daemon.

Jogando fora arquivos de log

Não recomendamos jogar fora todas as informações de registro. Instalações que estão sujeitas a problemas de segurança rotineiramente descobrem que os arquivos de log fornecem evidência importante de intrusão. Os arquivos de log também são importantes para ajudá-lo a alertar sobre problemas de hardware e de software. Em geral, dada uma quantidade de espaço em disco confortável, você deve manter os dados pelo menos por um mês. No mundo real pode levar um tempo até que você descubra que sua instalação foi invadida por um hacker e que você precisa rever os logs. Caso precise retornar ao passado, poderá recuperar arquivos de log antigos de suas fitas de backup.

Alguns administradores permitem que os arquivos de log cresçam até eles se tornarem incômodos, depois os reiniciam do zero. Esse plano é melhor do que não manter nenhum dado em si, porém ele não garante que as entradas de log serão retidas por qualquer período de tempo em particular. O uso médio de disco também pode ser maior do que com outros esquemas de gerenciamento.

www.snow666.blogspot.com
Snow666

Rodízio de arquivos log

Muitas instalações armazenam separadamente informações de log de cada dia, algumas vezes de uma maneira compactada. Arquivos diários são mantidos por um período de tempo específico e depois eliminados. É prático manter arquivos de log descompactados para que eles possam ser facilmente localizados através de **grep**. Em nossa instalação dedicamos uma partição do disco num host de registro central (**/var/log**) para arquivos de log. Compactamos dados que estão há mais de uma semana lá, por meio do comando **gzip**.

Uma maneira comum de preservar informações de log por um período fixo é conhecida como "rodízio". Num sistema de rodízio, mantemos arquivos de backup que têm um dia de idade, dois dias de idade e assim por diante. Cada dia, um script ou programa utilitário renomeia os arquivos para deslocar os de data mais antiga para o final da cadeia.

Se, por exemplo, um arquivo de log se chama **logfile**, as cópias de backup poderiam ser chamadas **logfile.1**, **logfile.2** e assim por diante. Se você mantiver dados de uma semana, haverá um **logfile.7**, mas nenhum **logfile.8**. Todo dia, os dados em **logfile.7** são perdidos já que **logfile.6** grava por cima dele.

Suponha que um arquivo necessite de atenção diária e você queira arquivar seu conteúdo por três dias (para fazer com que o exemplo fique curto). O script a seguir implementaria uma política de rodízio apropriada:

```
#!/bin/sh
cd /var/log
mv logfile.2 logfile.3
mv logfile.1 logfile.2
mv logfile logfile.1
cat /dev/null > logfile
chmod 600 logfile
```

A maioria das distribuições do Linux (inclusive todas as de nossos exemplos, exceto o SuSE) fornece um excelente utilitário de rodízio de logs chamado **logrotate**, que descrevemos a partir da seção 11.3. É muito mais fácil (e mais confiável) do que criar os seus próprios scripts e vale a pena procurá-lo e instalá-lo caso sua distribuição não o inclua.

Informações de propriedade são importantes para alguns arquivos de log. Talvez seja necessário executar seu script de rodízio por meio do **cron** como o proprietário dos arquivos de log e não como root ou quem sabe você precise adicionar um comando **chown** à seqüência.

Alguns sites identificam arquivos de log por data, e não pelo número de seqüência; por exemplo, **logfile.terca** ou **logfile.26agosto**. Esse sistema é um pouco mais difícil de ser implementado, porém pode valer a pena o esforço caso você se refira freqüentemente a arquivos de logs antigos. É muito mais fácil configurar em Perl do que em **sh**. Uma sintaxe útil que não requer qualquer programação é a seguinte:

```
mv logfile.`date +%Y.%m.%d`
```

Esse esquema apresenta a vantagem de fazer com que **ls** ordene os arquivos de log cronologicamente.

Alguns daemons mantêm seus próprios arquivos de log abertos todo o tempo, Devido à maneira pela qual o sistema de arquivos funciona, nosso script de exemplo não pode ser usado com tais daemons. Em vez de fluir para o **logfile** recriado, os dados de log continuarão a ir para **logfile.1**; a referência ativa ao arquivo original persiste mesmo depois de o arquivo ter sido renomeado. Para instalar um novo arquivo de log temos de sinalizar isto para o daemon ou então anular o processo e reiniciá-lo.

Eis um exemplo atualizado que usa tanto compactação quanto sinais:

```
#!/bin/sh
cd /var/log
mv logfile.2.gz logfile.3.gz
mv logfile.1.gz logfile.2.gz
mv logfile logfile.1
cat /dev/null > logfile
chmod 600 logfile
kill -signal pid
gzip logfile.1
```

signal representa o sinal apropriado para o programa gravar o arquivo de log; *pid* é o seu ID de processo. O sinal pode ser hardcode no script, porém você tem de determinar dinamicamente o PID do daemon seja lendo um arquivo que o daemon deixou para trás para você (por exemplo, **/var/run/syslog.pid**, descrito a seguir) seja usando a variante **killall** de **kill**, que é capaz de pesquisar o PID para você na tabela de processos. Por exemplo, o comando

www.snow666.blogspot.com
Snow666


```
killall -e -HUP syslogd
```

é equivalente a

```
kill -HUP `cat /var/run/syslog.pid`
```

Arquivando arquivos de log

Algumas instalações arquivam todos os dados de contabilidade e arquivos de log por uma questão de política, talvez para fornecer dados para uma possível auditoria. Nessa situação, os arquivos de log devem ser os primeiros a entrar no rodízio no disco e depois gravados em fita ou outra mídia permanente. Esse esquema reduz a freqüência dos backups de fita e nos dá rápido acesso a dados recentes.

Veja o Capítulo 10 para obter maiores informações sobre backups.

Os arquivos de log sempre devem ser incluídos como parte de sua seqüência regular de backup. Pelo fato de conterem informações que são vitais para investigar incidentes de segurança, eles devem ser colocados em backup com a freqüência mais alta que sua programação de dumps permitir. Os arquivos de log mudam freqüentemente, de modo que eles podem representar uma porção significativa de informações de sistema que são armazenadas em backups incrementais. Tenha em mente a interação entre sua política de registro e sua política de backups ao elaborar ambas.

Além de serem armazenados como parte de backups regulares, os logs também podem ser arquivados numa série de fitas distintas. Fitas distintas são mais enfadonhas, porém impõem menos em termos de trabalho de documentação e não interferirão na sua habilidade de reciclar fitas de dumps. Se você usar fitas distintas, sugerimos que use o formato **tar** e crie um script para automatizar seu esquema de backup.

11.2 ARQUIVOS DE LOG NO LINUX

Os sistemas UNIX tradicionais são normalmente criticados por suas inconsistências e até mesmo por sua abordagem um tanto bizarra ao processo de registro. Felizmente, os sistemas Linux são geralmente um pouco mais sensatos, embora cada distribuição possua sua maneira própria de nomear e de dividir os arquivos de log. Para a maior parte, os pacotes Linux enviam suas informações de registro para arquivos no diretório **/var/log**. Em algumas distribuições, os logs também são armazenados em **/var/adm**.

*O formato do arquivo **syslog.conf** é descrito na seção 11.4.*

A maioria dos programas nos dias de hoje na verdade envia suas entradas de log para um sistema central chamado syslog, que é descrito posteriormente neste capítulo. A configuração de syslog padrão faz, tipicamente, um dump da maioria dessas mensagens em algum ponto de **/var/log**. Verifique o arquivo de configuração de syslog, **/etc/syslog.conf**, para descobrir os detalhes.

A Tabela 11.1 compila informações sobre alguns dos arquivos de log mais comuns em nossas distribuições exemplificadas. Ela lista, especificamente:

- Os arquivos de log a serem arquivados, sintetizados ou truncados.
- O programa que cria cada um deles.
- Uma indicação de como cada nome de arquivo é especificado.
- A freqüência de atenção ou de limpeza que consideramos razoável.
- As distribuições (entre nossos exemplos) que usam o arquivo de log.
- Uma descrição do conteúdo do arquivo.

Os nomes de arquivos são relativos a **/var/log** a menos que indicado em contrário.

O caractere na coluna Onde informa como o arquivo de log é especificado: S para programas que usam syslog. F para programas que usam um arquivo de configuração e H se o nome de arquivo estiver definido diretamente em código. A coluna Freq indica a freqüência de limpeza que sugerimos.

Os arquivos de log pertencem geralmente à root, embora convenções para a propriedade e para o modo dos arquivos de log possam variar entre as distribuições. Em nossa opinião, a maioria dos logs deve ter atribuído a si o modo 600 (leitura e gravação somente para o proprietário), pois seu conteúdo tem potencial de utilidade para hackers. No mínimo, os arquivos **secure**, **auth.log** e **sudo.log** devem estar fora do alcance da navegação casual. Jamais dê permissões de gravação sobre qualquer arquivo de log a qualquer um, exceto o proprietário.

Vale a pena notar que muitos dos arquivos de log contidos na Tabela 11.1 são mantidos por syslog, mas a configuração de syslog padrão varia muito entre os sistemas. Com um arquivo **/etc/syslog.conf** mais consistente, os arquivos de log parecem relativamente similares entre as distribuições Linux.

www.snow666.blogspot.com
Snow666

Tabela 11.1 Arquivos de log na parada.

Arquivo	Programa	Onde[a]	Freq[a]	Distribs[a]	Conteúdo
messages	vários	S	W	todas	Normalmente o arquivo de log de sistema principal.
syslog	vários	S	W	D	Normalmente o arquivo de log de sistema principal.
auth.log	**su**, etc.[b]	S	M	DR	Autorizações.
secure	xinetd	S	M	R	Mensagens de autorização privadas.
boot.log	scripts **rc**	F[c]	M	R	Saída dos scripts de inicialização de sistema.
dmesg	kernel	H	_	RD	Dump do buffer de mensagens do kernel.
boot.msg	kernel	H	_	S	Dump do buffer de mensagens do kernel.
wtmp	**login**	H	M	todas	Registros de login (binários).
lastlog	**login**	H	_	todas	Último horário de login por usuário (binário).
setuid.*	**checksecurity**	H	D	D	Lista de arquivos setuid no sistema.
debug	vários	S	D	D	Saída de depuração (quando habilitada).
faillog	**login**	H	W	DS	Tentativas de login mal sucedidas.
sudo.log	**sudo**	F	W	_	Log do acesso root através de **sudo**.

a. Onde: S=Syslog, H=Definido no código, F=Arquivo de Configuração
Freq: D=Diária, W=Semanal, M=Mensal
Distrib: R=Red Hat, D=Debian, S=SuSE
b. **passwd**, **login** e **shutdown** também gravam no log de autorizações. Ele se encontra em **/var/adm** no Red Hat.
c. Na verdade efetua os logs através de syslog, porém o recurso e o nível são configurados em **/etc/initlog.conf**.

Arquivos de log especiais

A maioria dos logs são arquivos de texto para os quais linhas são escritas à medida que eventos "interessantes" vão ocorrendo. Entretanto, alguns dos logs apresentados na Tabela 11.1 possuem um contexto muito diferente.

/var/log/wtmp contém um registro dos logins e logouts de usuários, bem como entradas que indicam quando o sistema foi reinicializado ou desligado. Trata-se de um arquivo de log relativamente genérico, já que novas entradas são simplesmente adicionadas ao final do arquivo. Entretanto, o arquivo **wtmp** é mantido num formato binário. Use o comando **last** para decodificar as informações. Apesar de seu formato incomum, o arquivo **wtmp** deve ser colocado em rodízio ou truncado como qualquer outro arquivo de log, pois sua tendência natural é a de crescer indefinidamente.

Veja a nota de rodapé 4 do Capítulo 10 para obter maiores informações sobre arquivos esparsos.

/var/log/lastlog contém informações similares à **/var/log/wtmp**, porém ele registra somente o tempo de login de cada usuário. É um arquivo binário esparso que é indexado por UID. Ele permanecerá menor, caso seus UIDs sejam atribuídos em alguma forma de seqüência numérica, embora isto certamente não seja nada para se perder horas de sono na vida real. **lastlog** não precisa entrar em rodízio, pois seu tamanho permanece constante a menos que novos usuários façam o login.

O utilitário **checksecurity** do Debian é executado todo dia (por meio de **/etc/cron.daily/standard**) para enumerar o complemento de programas setuid do sistema. **checksecurity** grava quaisquer alterações relativas à referência anterior em **/var/log/setuid.changes**; ele também mantém as listas atual e anterior de programas setuid nos arquivos **setuid.today** e **setuid.yesterday**. Verifique o horário de modificação no arquivo **setuid.yesterday** para ver quando ocorreu a mudança mais recente.

Registro do horário de inicialização e do kernel

Os scripts de inicialização de sistema e de kernel apresentam alguns desafios especiais no domínio de registro. No caso do kernel, o problema é criar um registro permanente do processo de inicialização e a operação do kernel sem criar dependências em qualquer sistema de arquivos em particular ou organização de sistema de arquivos. No caso dos scripts de inicialização, o desafio é capturar uma narrativa coerente do procedimento de inicialização sem prender permanentemente qualquer um dos daemons de sistema a um arquivo de log de inicialização, interferindo com o próprio registro de qualquer programa ou arruinando os scripts de inicialização com entradas duplicadas ou redirecionamentos de saída.

O registro de kernel é conseguido fazendo com que o kernel armazene suas entradas de logs num buffer interno de tamanho limitado. O buffer é grande o suficiente para acomodar mensagens sobre todas as atividades do kernel no momento da inicialização. Assim que o sistema tiver inicializado, um processo de usuário pode acessar o

Snow666

buffer de log do kernel e fazer uma disposição final de seu conteúdo. As distribuições realizam isso tipicamente executando o comando **dmesg** e redirecionando sua saída para **/var/log/dmesg** (Red Hat e Debian) ou **/var/log/boot.msg** (SuSE). Esse é o melhor local para procurar informações sobre o processo de inicialização mais recente.

O registro do kernel, a partir de então, é tratado por um daemon chamado **klogd**. As funções de **klogd** são, na verdade, um superconjunto daquelas de **dmesg**; além de fazer o dump do log de kernel e sair, ele também pode ler mensagens do buffer de kernel à medida que são geradas e as repassar a um arquivo ou para o syslog. Em operação normal, o **klogd** executa este último modo; o syslog processa as mensagens de acordo com as instruções para o recurso "kern".

Os scripts de inicialização de nossos exemplos de distribuição não usam o flag **-c** de **dmesg** ao fazerem o dump inicial das mensagens de logs, de modo que o buffer de mensagens do kernel é lido mas não reinicializado e os submete ao syslog. Por essa razão, algumas entradas aparecem tanto no arquivo **dmesg** como **boot.msg** e num outro arquivo gerenciado por syslog como **/var/log/messages**.

Um outro problema no registro de kernel é o gerenciamento apropriado da console de sistema. À medida que o sistema está inicializando, é importante para todos que a saída vá para a console. Porém, assim que o sistema estiver no ar e ativo, as mensagens de console podem ser mais um distúrbio do que uma ajuda, particularmente se a console for usada para logins.

Tanto **dmesg** quanto **klogd** nos permitem configurar o nível de registro de console do kernel com um flag de linha de comando. Por exemplo:

```
% sudo dmesg -n 2
```

O nível 7 é o mais prolixo e inclui informações de depuração. O nível 1 inclui somente mensagens de pânico do kernel (os níveis de menor numeração são os mais severos. Todas as mensagens de kernel continuam indo para o buffer central (e para syslog) independentemente de serem encaminhadas para a console ou não.

O registro para os scripts de inicialização de sistema não é, infelizmente, tão bem gerenciado como o registro de kernel. O Red Hat usa um comando chamado **initlog** para capturar a saída dos comandos de inicialização e submetê-la ao syslog. Infelizmente, **initlog** tem de ser mencionado explicitamente sempre que um comando for executado, para que a captura de informações seja obtida, porém com um certo nível de dificuldade. As entradas eventualmente atingem seu objetivo em **/var/log/boot.log**.

Nossos outros sistemas de exemplo não fazem nenhum esforço coerente para capturar um histórico da saída dos scripts de inicialização. Algumas informações são colocadas em log por daemons e comandos individuais, mas grande parte não é registrada em log.

11.3 LOGROTATE: GERENCIANDO ARQUIVOS DE LOG

O excelente utilitário **logrotate** de Erik Troan implementa uma série de políticas de gerenciamento de logs e é padrão em sistemas Red Hat e Debian. Os usuários do SuSE devem, de uma vez, instalar o RPM.

O arquivo de configuração **logrotate** é formado por uma série de especificações para grupos de arquivos de log a serem gerenciados. Opções que aparecem fora do contexto da especificação de um arquivo de log (como `errors`, `rotate` e `weekly` no exemplo a seguir) aplicam-se a todas as especificações seguintes; elas podem ser sobrepostas dentro da especificação para um determinado arquivo de log e também podem ser especificadas novamente no arquivo para modificar os padrões.

Eis um exemplo um tanto inventado que manipula vários arquivos de log diferentes:

```
# Exemplo de política de rodízio de logs
errors sa-book@admin.com
rotate 5
weekly

/var/log/messages {
	postrotate
		/bin/kill -HUP `cat /var/run/syslog.pid`
	endscript
}

/var/log/samba/*.log {
	notifempty
	copytruncate
	sharescripts
```

Snow666


```
    postrotate
        /bin/kill -HUP `cat /var/lock/samba/*.pid`
    endscript
}
```

Essa configuração faz o rodízio de **/var/log/messages** toda semana. Ela mantém cinco versões do arquivo e notifica **syslogd** cada vez que o arquivo é reinicializado. Os arquivos de log do Samba (dos quais pode haver vários) também entram num rodízio num esquema semanal, porém em vez de serem colocados de lado e reinicializados, eles são copiados e depois truncados. Os daemons do Samba recebem sinais HUP somente depois de todos os arquivos de log terem passado por rodízio.

A Tabela 11.2 apresenta as opções mais úteis de **logrotate.conf**.

Tabela 11.2 Opções de logrotate.

Opção	Significado
`compress`	Compacta todas as versões não concorrentes do arquivo de log.
`daily, weekly, monthly`	Faz o rodízio de arquivos de log na programação especificada.
`delaycompress`	Compacta todas as versões, exceto a atual e a próxima mais recente.
`endscript`	Marca o final de um script `prerotate` ou `postrotate`.
`errors` *`emailaddr`*	Envia e-mails notificando erros para o endereço de e-mail (*`emailaddr`*) especificado.
`missingok`	Não reclama caso o arquivo de log não exista.
`notifempty`	Não faz o rodízio do arquivo de log caso ele esteja vazio.
`olddir` *`dir`*	Especifica que as versões mais antigas do arquivo de log devem ser colocadas em *dir*.
`postrotate`	Introduz um script para ser executado depois de o log ter passado pelo rodízio.
`prerotate`	Introduz um script para ser executado antes de quaisquer modificações serem feitas.
`rotate` *`n`*	Inclui *n* versões do log no esquema de rodízio.
`sharedscripts`	Executa somente scripts uma vez para todo o grupo de logs.
`size=`*`logsize`*	Faz o rodízio caso o tamanho do arquivo de log seja maior que *logsize* (por exemplo, 100K, 4M).

logrotate normalmente é executado por meio de **cron** uma vez por dia. Seu arquivo de configuração é **/etc/logrotate.conf**, porém vários arquivos de configuração (ou diretórios contendo arquivos de configuração) podem aparecer na linha de comando **logrotate** de uma vez. Esse recurso é usado com grande efeito por Red Hat e Debian que definem o diretório **/etc/logrotate.d** como um local-padrão para arquivos de configuração **logrotate**. Pacotes de software capazes de reconhecer **logrotate** (dos quais há vários) são capazes de dar instruções de gerenciamento de log como parte de seu procedimento de instalação, simplificando em muito a administração.

11.4 Syslog: O Registrador de Eventos do Sistema

Syslog, originalmente escrito por Eric Allman, é um sistema de registro abrangente. Ele possui duas funções importantes: liberar os programadores da mecânica entediante da gravação de arquivos de log e colocar os administradores no controle do processo de registro. Antes de syslog, todo programa era livre para praticar sua própria política de registro. Administradores de sistemas não tinham nenhum controle sobre quais informações eram preservadas ou onde eram armazenadas.

Syslog é bastante flexível. Ele permite que mensagens sejam ordenadas pela fonte e importância ("nível de gravidade") e direcionadas para uma série de destinos: arquivos de log, terminais de usuários ou até mesmo outras máquinas. A habilidade de syslog de centralizar o registro para uma rede é um dos seus vários recursos valiosos.

Syslog é formado por três partes:

- **syslogd**, o daemon de registro (juntamente com seu arquivo de configuração, **/etc/syslog.conf**)
- **openlog** et al., rotinas de bibliotecas que submetem mensagens a **syslogd**
- **logger**, um comando em nível de usuário que submete entradas de log do shell

Na discussão a seguir, cobrimos primeiramente a configuração de **syslogd** e depois mostramos brevemente como usar syslog de scripts Perl.

syslogd é disparado no momento do boot e é executado continuamente; ele não pode ser gerenciado com **inetd**. Programas que reconhecem syslog gravam entradas de log (usando a rotina de bibliotecas **syslog**) no arquivo

www.snow666.blogspot.com
Snow666

especial **/dev/log**, um soquete de domínio UNIX. **syslogd** lê mensagens de **/dev/log**, consulta seu arquivo de configuração e despacha cada mensagem para os destinos apropriados.

O daemon de registro de kernel, **klogd**, é responsável pela recuperação de mensagens do buffer de log interno do kernel e pelo encaminhamento delas para **syslogd**.

Um sinal de suspensão (HUP, sinal 1) faz com que **syslogd** feche seus arquivos de log, releia seu arquivo de configuração e inicie o registro novamente. Se você modificar **/etc/syslog.conf**, terá de enviar um sinal de suspensão para **syslog** fazer com que suas alterações entrem em efeito. Um sinal TERM faz com que **syslogd** saia.

syslogd grava seu ID de processo para o arquivo **/var/run/syslogd.pid**. Essa convenção facilita o envio de sinais para **syslogd** por meio de um script. Por exemplo, o comando a seguir envia um sinal de suspensão:

```
# kill -HUP `/bin/cat /var/run/syslogd.pid`
```

Tentar compactar ou modificar um arquivo de log que **syslogd** tenha aberto para gravação não é saudável e possui resultados imprevisíveis. O procedimento apropriado é deixar o log antigo de lado, recriar o log com as mesmas propriedades e permissões e depois enviar um sinal HUP para **syslogd**. Esse procedimento é facilmente implementando por meio de **logrotate**; veja um exemplo na seção 11.3.

Configurando syslogd

O arquivo de configuração **/etc/syslog.conf** controla o comportamento de **syslogd**. Trata-se de um arquivo de texto com um formato relativamente simples. Linhas em branco e linhas com um sinal # na primeira coluna são ignoradas. O formato básico é:

```
selector <Tab> action
```

Por exemplo, a linha

```
mail.info     /var/log/maillog
```

faria com que as mensagens do sistema de e-mails fossem salvas no arquivo **/var/log/maillog**.

Seletores identificam o programa ("recurso") que está enviando uma mensagem de log e o grau de gravidade da mensagem com a sintaxe *recurso.nível*. Tanto os nomes dos recursos quanto os níveis de gravidade têm de ser escolhidos de uma lista reduzida de valores definidos; programas não podem fazer isso por conta própria. Os recursos são definidos para o kernel, para grupos de utilitários comuns e para programas escritos localmente. Tudo o mais é classificado sob o recurso genérico "user".

Os seletores podem conter as palavras-chave especiais * e `none`, significando, respectivamente, tudo ou nada. Um seletor pode incluir vários recursos separados por vírgulas. Vários seletores podem também ser combinados com ponto-e-vírgulas.

Em geral, os seletores sofrem a ação da operação lógica OR; uma mensagem atendendo ao critério de qualquer seletor estará sujeita à *ação* da linha. Entretanto, um seletor com um nível `none` exclui os recursos listados independentemente do que outros seletores na mesma linha possam dizer.

Eis alguns exemplos de maneiras para formatar e combinar seletores:

```
recurso.nivel                    acao
recurso1,recurso2.nivel          acao
recurso1.nivel1,recurso2.nivel2  acao
*.nivel                          acao
*.nivel;recursoruim.none         acao
```

A Tabela 11.3 lista os nomes de recursos de syslog válidos.

Tabela 11.3 Nomes de recursos de syslog.

Recurso	Programas que o utilizam
*	Todos os recursos, exceto "mark"
auth	Comandos relacionados com segurança e autorização
authpriv	Mensagens de autorização confidenciais/privadas[a]
cron	O daemon **cron**
daemon	Daemons de sistema
ftp	O daemon FTP, **ftpd**

Snow666

Tabela 11.3 Nomes de recursos de syslog. (*continuação*)

Recurso	Programas que o utilizam
kern	O kernel
local0-7	Oito variantes de mensagem local
lpr	O sistema de spooling de impressora de linha
mail	**sendmail** e outros programas relacionados com e-mail
mark	Timestamps gerados em intervalos regulares
news	O sistema de notícias Usenet
syslog	Mensagens internas de **syslogd**
user	Processos de usuário (o padrão se não especificado)
uucp	Reservado para UUCP, que não o utiliza

a. *Todas* as mensagens relacionadas com autorização são confidenciais. Nem as mensagens authpriv nem as mensagens auth devem ter autorização para leitura para todo o mundo.

syslogd em si produz mensagens de timestamp que são logadas caso o recurso "mark" apareça em **syslog.conf** para especificar um destino para elas. Os timestamps podem ajudá-lo a descobrir que sua máquina travou entre 3h00 e 3h20 e não somente "em algum momento durante a última noite". Essas informações podem ser de grande ajuda quando estivermos depurando problemas que pareçam ocorrer regularmente. Por exemplo, muitas instalações passaram por paradas misteriosas quando o pessoal de limpeza ligava aspiradores de pó durante a madrugada, acionando os disjuntores.

Se seu sistema estiver bem ocupado, outras mensagens de log normalmente fornecerão informações de timestamp adequadas. Porém nas primeiras horas da madrugada, isso nem sempre é o caso.

A Tabela 11.4 apresenta os níveis de gravidade do syslog em ordem decrescente de importância.

Tabela 11.4 Níveis de gravidade de syslog (em ordem decrescente em termos de gravidade).

Nível	Significado aproximado
emerg	Situações de pânico
alert	Situações urgentes
crit	Condições críticas
err	Outras condições de erros
warning	Mensagens de alerta
notice	Coisas que podem merecer investigação
info	Mensagens informativas
debug	Somente para depuração

O grau de gravidade de uma mensagem especifica sua importância. No arquivo **syslog.conf**, os níveis indicam a importância *mínima* que uma mensagem deve ter para merecer ser registrada em log. Por exemplo, uma mensagem do sistema de e-mail no nível de alerta concordaria com o seletor `mail.warning`, bem como com os seletores `mail.notice`, `mail.info`, `mail.debug`, `*.warning`, `*.notice`, `*.info` e `*.debug`. Se **syslog.conf** especifica que mensagens `mail.info` devem ser logadas num arquivo, então mensagens `mail.warning` também irão para lá.

Como refinamento da sintaxe básica, a versão Linux de syslog também permite que os caracteres = e ! sejam antecedidos aos níveis de prioridade para indicarem "somente esta prioridade" e "exceto esta prioridade e maiores". A Tabela 11.5 mostra alguns exemplos.

Tabela 11.5 Exemplos de qualificadores de nível de prioridade em syslog.conf.

Seletor	Significado
`mail.info`	Seleciona mensagens relacionadas a e-mail de prioridade info ou superior
`mail.=info`	Seleciona somente mensagens de prioridade info
`mail.info;mail.!err`	Seleciona somente as prioridades info, notice e warning
`mail.debug;mail.!=warning`	Seleciona todas as prioridades, exceto warning

Snow666

O campo *action* informa o que fazer em um mensagem. A Tabela 11.6 enumera suas opções.

Tabela 11.6 Ações de syslog.

Ação	Significado
filename	Grava a mensagem em um arquivo na máquina local
@hostname	Encaminha a mensagem para o **syslogd** em *hostname* (nome do host)
@ipaddress	Encaminha a mensagem para o host no endereço IP *ipaddress*
|fifoname	Escreve a mensagem para o pipe com nome *fifoname*[a]
user1,user2,...	Escreve a mensagem nas telas dos usuários caso eles estejam logados
*	Escreve a mensagem para todos os usuários que estivem logados no momento

a. Veja **info mkfifo** para obter maiores informações.

Consulte o Capítulo 18 para obter maiores informações sobre como os nomes de hosts são traduzidos em endereços IP.

Se uma ação *filename* (ou *fifoname*) for usada, o nome deve ser um caminho absoluto. Se especificarmos um nome de arquivo inexistente, **syslogd** criará o arquivo quando uma mensagem for direcionada pela primeira vez a ele.[1] Podemos anteceder uma ação *filename* por um traço para indicar que o sistema de arquivos não deve ser sincronizado (**sync**) depois de cada entrada de log ter sido registrada. **sync** ajuda a preservar o máximo possível de informações de log no caso de uma pane, porém ele é custoso em termos de desempenho de disco.

Se for usado um *hostname* em vez de um endereço IP, ele tem de ser convertido por meio de um mecanismo tradutor como DNS ou NIS.

Embora possamos usar vários recursos e níveis num seletor, não há nenhuma opção para ações múltiplas. Para enviarmos uma mensagem a dois lugares (como, por exemplo, para um arquivo local e para um host de registro central), podemos incluir no arquivo de configuração duas linhas com os mesmos seletores.

Pelo fato de as mensagens de syslog poderem ser usadas para montar um ataque de recusa de serviços, **syslogd** não aceitará mensagens de log de outras máquinas a menos que seja iniciado com o flag **-r**. Como padrão, **syslogd** também se recusa a agir como um encaminhador de mensagens de terceiros; mensagens provenientes de um host de rede não podem ser encaminhadas para um outro. Use o flag **-h** para anular esse padrão. (Se quiser que essas opções estejam ativas todo o tempo, adicione flags ao script **/etc/rc.d/init.d/syslog** no Red Hat e **/etc/init.d/syslogd** no Debian. O Linux SuSE fornece uma sintaxe mais suave e somente requer que modifiquemos a variável SYSLOGD_PARAMS definida em **/etc/rc.config**.)

Exemplos de arquivos de configuração

A configuração de syslog é uma área na qual as distribuições Linux variam muito. Já que é relativamente fácil lermos um arquivo **syslog.conf**, não revisaremos detalhadamente os arquivos de configuração de nossos exemplos de distribuição; eles são bem diretos. Em vez disso, veremos algumas maneiras que poderíamos usar para configurar o registro caso optemos por partir do padrão do sistema.

A seguir apresentamos três exemplos de arquivos de configuração **syslog.conf** que correspondem, respectivamente, a uma máquina dedicada numa pequena rede, uma máquina-cliente numa rede maior e um host de registro central na mesma rede grande. O host de registro central é chamado de "netloghost".[2]

Máquina dedicada

Uma configuração básica para uma máquina dedicada é apresentada a seguir:

```
# Arquivo syslog.conf para máquina dedicada ou pequena rede

# emergências: informe a todo o mundo quem está logado em
*.emerg                                    *
```

1. Note que esse comportamento é o contrário daquele da implementação original de syslog, que exigia que arquivos de log fossem criados antecipadamente.
2. Mais precisamente, ele usa "netloghost" como um de seus nomes alternativos para nome de host. Isso permite que a identidade do host de logs seja modificada com um nível menor de reconfiguração. Um nome alternativo pode ser adicionado em **/etc/hosts** ou configurado com um registro CNAME em DNS. Veja a página 313 para obter maiores informações sobre registros CNAME DNS.

Snow666


```
# mensagens importantes

*.warning;daemon,auth.info,user.none /var/log/messages
# erros de impressão
lpr.debug                            /var/log/lpd-errs
```

A primeira linha não comentada envia mensagens de emergência às telas de todos os usuários atuais. Um exemplo de mensagens de nível de emergência é aquela gerada por **shutdown** quando o sistema está prestes a ser desligado.

A segunda linha envia mensagens importantes para **/var/log/messages**. O nível info está abaixo de warning, de modo que a instrução `daemon,auth.info` inclui registro adicional de **passwd**, **su** e programas daemon. A terceira linha envia mensagens de erro de impressão para **/var/log/lpd-errs**.

Cliente em rede

Um cliente em rede tipicamente encaminha mensagens críticas para uma máquina central de registro:

```
# Arquivo syslog.conf do Departamento de Computação para
# máquinas não mestras

# Emergências: informe todo o mundo que se encontra logado sobre
*.emerg;user.none                    *

# Encaminhe mensagens importantes para o logger central
*.warning;lpr,local1.none            @netloghost
daemon,auth.info                     @netloghost

# Também encaminhe material local para o logger central
local2.info;local0,local7.debug      @netloghost

# logs cardd através do recurso local1 -- envie para boulder
local1.debug                         @boulder.colorado.edu

# Mantenha os erros de impressão localmente
lpr.debug                            /var/log/lpd-errs

# logs sudo para local2 – mantenha uma cópia aqui também
local2.info                          /var/log/sudo.log

# Mantenha as mensagens de kernel localmente
kern.info                            /var/log/kern.log
```

Esta configuração não mantém localmente muita informação de log. Vale a pena citar que se netloghost estiver inativo ou não puder ser alcançado, as mensagens de log serão irremediavelmente perdidas. Talvez você queira manter algumas duplicatas locais adicionais de mensagens importantes para se precaver contra essa possibilidade.

Numa instalação com software local instalado, um grande número de mensagens pode ser registrado inapropriadamente para o recurso user, nível emerg. Neste exemplo, user/emerg foi excluído especificamente com a instrução `user.none` da primeira linha.

A segunda e terceira linhas encaminham todas as mensagens importantes para o host de registro central; mensagens do sistema de impressão e do sistema de acesso a cartão englobando toda a instalação são excluídas explicitamente. A quarta linha encaminha informações de registro locais também para netloghost. A quinta linha encaminha informações de registro de cartão para o host de registro abrangendo toda a instalação, boulder. As duas últimas entradas mantêm cópias locais dos erros de impressão e mensagens de log **sudo**.

Host central para registro

Este exemplo é para netloghost, o host centralizado e seguro para registro para uma rede de tamanho razoável com cerca de 7.000 hosts.

```
# Arquivo syslog.conf do Departamento de Computação, host
# para registro mestre

# Emergências para a console e arquivo de log, com marcas de horário
*.emerg                                  /dev/console
```

Snow666


```
*.err;kern,mark.debug;auth.notice                /dev/console
*.err;kern,mark.debug;user.none                  /var/log/console.log
auth.notice                                      /var/log/console.log

# Encaminhe mensagens que não são de emergência para os arquivos de log
# usuais
*.err;user.none;kern.debug                       /var/log/messages
daemon,auth.notice;mail.crit                     /var/log/messages
lpr.debug                                        /var/log/lpd-errs
mail.debug                                       /var/log/mail.log

# Material de autorização local como sudo e npasswd
local2.debug                                     /var/log/sudo.log
local2.alert                                     /var/log/sudo-errs.log
auth.info                                        /var/log/auth.log

# Outros materiais locais
local4.notice                                    /var/log/da.log
local7.debug                                     /var/log/tcp.log

# Material de usuário (o padrão caso não seja especificado nenhum
# recurso)
user.info                                        /var/log/user.log
```

Dados de log provenientes de programas locais e de **syslogd**s na rede são gravados em arquivos. Em alguns casos, a saída de cada recurso é colocada em seu próprio arquivo.

O host central para registro gera o time stamp para cada mensagem quando ele grava a mensagem. Os time stamps não refletem o horário no host originário. Se você tiver máquinas em vários fusos horários ou os relógios de seu sistema não estiverem sincronizados, os time stamps poderão ser um tanto incorretos.

Exemplo de saída de syslog

A seguir temos um pequeno trecho de um dos arquivos de log do host de syslog mestre do Departamento de Computação da Universidade de Colorado. Cerca de 200 hosts geram log para esta máquina.

```
Dec 18 15:12:42 av18.cs.colorado.edu sbatchd[495]: sbatchd/main: ls_info()
   failed: LIM is down; ty later; trying …
Dec 18 15:14:28 proxy-1.cs.colorado.edu pop-proxy[27283]: Connection from
   128.138.198.84
Dec 18 15:14:30 mroe.cs.colorado.edu pingem[271]: maltese-
   office.cs.colorado.edu has not answered 42 times
Dec 14 15:14:51 coyote.cs.colorado.edu PAM_unix[17405]: (sshd) session closed
   for user trent
Dec 18 15:15:48 proxy-1.cs.colorado.edu pop-proxy[27285]: Connection from
    12.2.209.183
Dec 18 15:15:50 av18.cs.colorado.edu last message repeated 100 times
```

Este exemplo contém entradas de vários hosts distintos (av18, proxy-1, coyote e outros) e de diversos programas: **sbatchd**, **pop-proxy**, **pingem** e da biblioteca Pluggable Authentication Modules.

Note a última linha do trecho, que reclama de uma mensagem que foi repetida 100 vezes. Para ajudar a manter os logs menores, syslog geralmente tenta aglutinar mensagens duplicadas e substituí-las por esse tipo de resumo. Entretanto, a máquina da qual este exemplo foi extraído aceita entradas de muitos outros hosts, de modo que esta mensagem em particular é um tanto enganadora. Ela se refere, na verdade, à entrada de log anterior a av18, não à entrada que a precede imediatamente no log composto.

É uma boa idéia percorrer seus arquivos de log regularmente. Determine o que é normal para que ao ocorrer uma anomalia, você possa reconhecê-la. Melhor ainda, configure um pós-processador de logs como **swatch** para capturar esses casos automaticamente; veja a seção 11.5.

Snow666

Desenvolvendo um esquema de registro em sua instalação

Numa instalação pequena é adequado configurar o registro de modo que erros de sistema e alertas importantes sejam mantidos num arquivo em cada máquina, de maneira muito parecida com o que era feito antes de termos o syslog. O arquivo **syslog.conf** pode ser personalizado para cada host.

Em uma grande rede, o registro centralizado é fundamental. Ele mantém o fluxo de informações de uma forma controlada e, com sorte, torna os dados de auditoria indisponíveis para uma pessoa que viola a segurança de uma máquina da rede. Os hackers normalmente editam logs de sistema para encobrir seus vestígios; se informações de log forem remetidas tão logo sejam geradas, será muito mais difícil de destruir. Porém esteja ciente de que qualquer um pode chamar **syslog** e falsificar entradas de log por meio de qualquer daemon ou utilitário. Syslog também usa o protocolo UDP, que não é garantidamente confiável; mensagens podem ser perdidas. Seu firewall não deveria permitir que instalações externas submetessem mensagens ao seu **syslogd**.

Veja o Capítulo 18 para obter maiores informações sobre distribuição de arquivos em uma rede.

Selecione uma máquina estável como seu servidor de registro, preferencialmente uma que esteja bem segura e não tenha muitos logins. Outras máquinas podem usar um arquivo de configuração genérico mantido num local centralizado. Portanto, somente duas versões do arquivo **syslogd.conf** precisam ser mantidas. Essa abordagem permite que o registro seja completo, porém ao mesmo tempo não se torna um pesadelo para o administrador.

Algumas instalações muito grandes podem querer adicionar mais níveis à hierarquia de registro. Infelizmente, o syslog retém o nome do host originário somente através de um host (hop). Se o host "cliente" enviar algumas entradas de log ao host "servidor", que as envia ao host "mestre", o mestre interpretará os dados como provenientes do servidor, e não do cliente.

Software que usa syslog

A Tabela 11.7 apresenta alguns dos programas que usam syslog, os recursos e níveis para onde são logados e uma breve descrição de cada programa. Algumas linhas são sombreadas para aumentar a legibilidade.

Tabela 11.7 Software que usa syslog.

Programa	Recurso	Níveis	Descrição
cron	cron	info	Daemon de agendamento de tarefa do sistema
ftpd	ftp	debug-crit	Daemon FTP (**wu-ftpd**)
imapd	mail	info-alert	Servidor de e-mail IMAP
inetd	daemon	err, warning	Superdaemon Internet
login	authpriv	info-err	Programas de login
lpd	lpr	info-err	Daemon de impressora de linha
named	daemon	info-err	Servidor de nomes (DNS)
passwd	auth	notice, warning	Programa de configuração de senhas
popper	local0	debug, notice	Servidor de e-mail POP3
sendmail	mail	debug-alert	Sistema de transporte de e-mails
shutdown	auth	notice	Pára o sistema
su	auth	notice	Comuta UIDs
sudo	local2	notice, alert	Programa **su** limitado
syslogd	syslog, mark	info-err	Erros internos, time stamps
tcpd	local7	debug-err	Envoltório TCP para **inetd**
vmlinuz	kern	todos	O kernel
xinetd	*configurável*	info (padrão)	Variante de **inetd** (Red Hat)

Com todas essas informações, deve ficar perfeitamente claro quais mensagens manter e quais descartar, não é mesmo? Bem, pode ser que não. Na prática, temos de saber somente quais são os níveis de registro úteis para o nosso sistema. É melhor começar com um nível de registro excessivo e, gradualmente, separar o joio do trigo, ou seja, eliminar os casos que não desejamos. Pare de eliminar quando se sentir confortável com a taxa de dados média.

Snow666

Depurando o syslog

O comando **logger** é útil para submetermos entradas de log por meio de scripts de shell. Também podemos usá-lo para testar alterações no arquivo de configuração de **syslogd**. Se, por exemplo, tivermos somente adicionado a linha

```
local5.warning        /tmp/evi.log
```

e quisermos verificar se ela está funcionando, devemos executar

```
% logger -p local5.warning "mensagem de teste"
```

Uma linha contendo "mensagem de teste" deve ser gravada para **/tmp/evi.log**. Se isso não acontecer, talvez você tenha esquecido de enviar um sinal de suspensão (HUP) a **syslogd**.

Tome cuidado ao fazer registro para o dispositivo-console, **/dev/console** ou para qualquer pseudoterminal ou porta que suporte controle de fluxo. Se alguém inadvertidamente digitar <Control-S> na console, a saída para a console será interrompida. Cada chamada a **syslog** será bloqueada e seu sistema ficará lento como uma lesma. Uma boa maneira de verificar esta condição de degeneração é enviar uma mensagem de syslog para a console com **logger**. Se **logger** ficar suspenso, você precisará descobrir qual é a porta problemática, digitar um <Control-Q> e repensar sua estratégia de registro.

Um outro inconveniente para jogar o registro para a console é que o fluxo de mensagens provocado por um grande problema pode tornar a console inutilizável precisamente no momento em que ela é mais necessária. Dependendo de como sua console é configurada e gerenciada (por exemplo, por meio de um servidor de console), o registro em console também pode apresentar algumas implicações em termos de segurança.

Utilizando syslog por intermédio de programas

As rotinas de biblioteca **openlog**, **syslog** e **closelog** permitem que os programas usem o sistema de syslog. Versões dessas rotinas de biblioteca se encontram disponíveis tanto para C quanto para Perl; descrevemos aqui somente a interface para Perl. Para importar as definições das rotinas de biblioteca, inclua a linha a seguir

```
use Sys::Syslog;
```

no início de seu script Perl.

A rotina **openlog** inicializa o registro em log, usando o nome do recurso especificado:

```
openlog(ident, logopt, facility);
```

As mensagens são colocadas em log com as opções especificadas por *logopt* e começam com a string de identificação *ident*. Se **openlog** não for usado, *ident* usa como padrão o nome do usuário atual, *logopt* usa uma string vazia e *facility* usa "user". A string *logopt* deve conter uma lista de opções separadas por vírgulas como indicada na Tabela 11.8.

Tabela 11.8 Opções de registro para a rotina openlog.

Opção	Significado
pid	Inclui o PID do processo atual em cada mensagem de log.
ndelay	Conecta-se imediatamente a **syslogd** (não espera uma mensagem ser submetida).
cons	Envia mensagens à console do sistema caso **syslogd** não possa ser alcançado.
nowait	Não aguarda [**wait**(3)] processos-filho ramificados enviarem mensagens para a console.

Por exemplo, uma chamada razoável a **openlog** poderia ser a seguinte

```
openlog("adminscript", "pid,cons", "local4");
```

A rotina **syslog** envia uma mensagem (*message*) a **syslogd**, que a registra em log na prioridade (*priority*) especificada:

```
syslog(priority, message, ...);
```

A data, o horário, o nome de host e a string *ident* da chamada **openlog** são anexadas à mensagem no arquivo de log. *message* pode ser seguida por vários outros parâmetros para formar uma especificação de saída estilo **printf** que pode incluir texto e o conteúdo de outras variáveis como, por exemplo:

```
syslog("info", "Entrega para '%s%' falhou após %d tentativas.", $user, $nAttempts);
```

Snow666

O símbolo especial `%m` expande-se para uma mensagem de erro derivada do valor atual de **errno** (o código de erro mais recente).

Uma string de prioridade do tipo *"level|facility"* configura tanto o nível de gravidade quanto o nome do recurso. Se não chamarmos **openlog** e especificarmos uma string *ident*, a rotina **syslog** também verificará se nossa mensagem (*message*) possui o formato de uma mensagem de erro UNIX padrão como

```
adminscript: User "nobody" not found in /etc/passwd file.
```

Se ela tiver, a parte anterior aos dois pontos será adotada secretamente como nossa string *ident*. Esses recursos úteis (porém não documentados) tornam desnecessário chamar **openlog** em si; porém, mesmo assim é uma boa idéia fazê-lo. É melhor especificar o nome do recurso em um lugar (a chamada **openlog**) do que repeti-lo ao longo de todo o código.

A rotina **closelog** fecha o canal de registro:

```
closelog();
```

Temos de chamar essa rotina caso queiramos reabrir o canal de registro com opções diferentes. É aconselhável chamar **closelog** quando nosso programa sai, porém fazer isto não é estritamente necessário.

Eis um exemplo completo:

```
use Sys::Syslog;

openlog("adminscript", "cons,pid",  "user");
syslog("warning","Aqueles a quem os deuses destruiriam, eles ensinam Basic em primeiro lugar.");
closelog();
```

Esse pequeno trecho de script gera a seguinte entrada de log (191 é o PID de **adminscript**):

```
Dec 28 22:56:24 moet.colorado.edu adminscript[191]: Aqueles a quem os deuses destruiriam, eles ensinam
    Basic em primeiro lugar.
```

11.5 Condensando Arquivos de Log em Informações Úteis

Syslog é excelente para ordenar e encaminhar mensagens de log, porém quando tudo estiver dito e feito, seu produto final ainda será um monte de arquivos de log. Embora eles possam conter todos os tipos de informações úteis, estes arquivos não irão ao seu encontro e o localizarão quando algo der errado. Uma outra camada de software é necessária para analisarmos os logs e certificar-nos de que as mensagens importantes não fiquem perdidas no meio de todo aquele palavrório.

Uma série de ferramentas gratuitas estão disponíveis para preencher esse nicho, e a maioria delas é bastante similar: elas varrem entradas de log recentes, as comparam com um banco de dados de expressões regulares e processam as mensagens importantes de alguma maneira para chamar a atenção. Algumas ferramentas enviam um relatório via e-mail para nós; outras podem ser configuradas para alertar-nos de forma auditiva, para imprimirem entradas de log em cores diferentes ou enviarem mensagens via pager. As ferramentas diferem basicamente no grau de flexibilidade e no tamanho de seus bancos de dados de padrões.

Dois dos pós-processadores de logs mais comumente usados são o **swatch** de Todd Atkins e **logcheck** de Craig Rowland[3]. **swatch** pode ser encontrado em

http://www.oit.ucsb.edu/~eta/swatch

e **logcheck** em

http://www.psionic.com/abacus/logcheck
http://doug.hunley.homeip.net/tools/

swatch é um script em Perl que recebe suas ordens por intermédio de um arquivo de configuração. A sintaxe de configuração é relativamente flexível e também fornece acesso aos "poderes mágicos" de comparação de padrões completos do Perl. Embora **swatch** seja capaz de processar um arquivo inteiro de uma só vez, ele se destina basicamente a ser deixado em execução para que possa rever mensagens novas à medida que elas forem chegando, tipo **tail -f**.

3. A última versão liberada por Craig Rowloand foi a 1.1 em junho de 1997. O site original do projeto era `http://www.psionic.com/abacus/logcheck`. Doug Hunley tem mantido uma versão com novas modificações em seu site http://doug.hunley.homeip.net/tools/. A versão Debian do **logcheck** é mantida independentemente das versões de Craig e do Doug. O mantenedor atual (outubro de 2003) é Steven Kemp. (N.R.T.)

www.snow666.blogspot.com
Snow666

Uma desvantagem de **swatch** é que você tem de criar sua configuração própria praticamente do zero; ele não reconhece sistemas específicos e as mensagens de log em si que eles podem gerar.

logcheck é um script mais básico escrito em **sh**; a distribuição também inclui um programa em C que **logcheck** usa para ajudá-lo a registrar sua posição dentro de um arquivo log. Já que **logcheck** sabe até que ponto ele leu um arquivo de log, provavelmente há menos chances de pular uma mensagem durante o momento da inicialização ou do desligamento; além disso, **logcheck** pode ser executado em intervalos por meio de **cron** em vez de ficar em execução continuamente.

logcheck vem acompanhado de modelos de bancos de dados para várias versões diferentes do UNIX (inclusive o Linux). Mesmo que não queira usar o script em si, vale a pena dar uma olhada nos padrões para ver se há algum que possa lhe servir.

Ambas as ferramentas possuem a desvantagem de operarem somente num único arquivo de log por vez. Se a sua configuração de syslog ordena mensagens em muitos arquivos diferentes, talvez você irá querer duplicar algumas das mensagens num arquivo centralizado que seja freqüentemente truncado, e então usar esse arquivo-sumário para alimentar um script de pós-processamento. É mais fácil do que configurar uma rede de scripts complicados para tratar de vários arquivos.

Uma outra ferramenta que vale a pena mencionar é **logwatch** de Kirk Bauer. Ela é mais um sintetizador de logs do que uma ferramenta para alerta e monitoramento contínuo, porém possui as vantagens de ser relativamente simples e de estar disponível na forma de um pacote RPM. Você poderá fazer o seu download em redhat.com ou então em

http://www.kaybee.org/~kirk/html/linux.html

Independentemente da ferramenta que você escolher para varrer arquivos de log, há uma série de coisas que deve procurar obrigatoriamente e trazer imediatamente ao conhecimento de um administrador:

- A maioria das mensagens relacionadas com segurança deve receber uma revisão imediata. Normalmente é útil monitorar logins com falha, tentativas **su** e **sudo** para capturar invasões em potencial antes que realmente aconteçcam. Se alguém simplesmente esquecer sua senha (como normalmente é o caso), uma oferta imediata e preventiva de ajuda dará uma boa impressão e solidificará sua reputação de seriedade.
- Mensagens sobre discos que foram preenchidos totalmente devem ser sinalizadas e devem ter uma intervenção imediata. Discos cheios normalmente fazem com que todo o trabalho útil seja paralisado.
- Mensagens que são repetidas muitas vezes merecem atenção, mesmo que somente em nome da organização.

11.6 EXERCÍCIOS

E11.1 Quais são as principais razões para se preservar arquivos de log antigos?

E11.2 Qual é a diferença entre **lastlog** e **wtmp**? Qual é uma política de rodízio razoável para cada um deles?

E11.3 Analise e entenda a seguinte linha de **syslog.conf**:

```
*.notice;kern.debug;lpr.info;mail.crit;news.err        /var/log/messages
```

Ela parece razoável?

E11.4 Execute um sintetizador de arquivos de log no maior arquivo de log de seu sistema. Ele revelou alguma coisa interessante? (Talvez exija acesso root.)

E11.5 Procure em seus arquivos de log entradas de **named**, o servidor de nomes DNS. Alguma máquina está tentando atualizar seus arquivos de domínio dinamicamente? Eles estão sendo bem-sucedidos em seu intento? (Talvez exija acesso root.)

★ **E11.6** Onde você encontraria o log de boot para sua máquina? Quais são as questões que afetam o registro no momento do boot? **klogd** resolve essas questões?

★ **E11.7** Investigue a política de registro em uso em sua instalação, inclusive a política de rodízio de arquivos de log. Quanto espaço em disco é dedicado ao registro? Por quanto tempo os arquivos de log são preservados? Você é capaz de antever circunstâncias na qual a política aplicada em sua instalação não seria adequada? Que solução você recomendaria? (Exige acesso root.)

★ **E11.8** Algumas mensagens de log são extremamente importantes e devem ser analisadas imediatamente por um administrador. Que sistema você poderia configurar para garantir que isso aconteça o mais rápido possível?

★ **E11.9** Escreva um programa em C ou script em Perl que submeta mensagens ao syslog com o recurso "user". (Talvez exija acesso root.)

Snow666

12 Drivers e o Kernel

O kernel é a parte do sistema responsável por ocultar o hardware do sistema sob uma interface de programação abstrata e de alto nível. Ele fornece muitos dos recursos que os usuários e os programas em nível de usuário tomam como certos. Por exemplo, o kernel monta todos os conceitos a seguir por meio de recursos de hardware de nível mais baixo:

- Processos (compartilhamento de tempo, espaços de endereços protegidos)
- Sinais e semáforos
- Memória virtual (swapping, paginação, mapeamento)
- O sistema de arquivos (arquivos, diretórios e espaço de nomes)
- Comunicação entre processos (pipes e conexões de rede)

O kernel contém drivers de dispositivo que gerenciam sua interação com peças de hardware específicas; o restante do kernel é, em grande parte, independente de dispositivo. O relacionamento entre o kernel e seus drivers de dispositivo é similar ao relacionamento entre processos em nível de usuário e o kernel. Quando um processo solicita ao kernel para "Ler os primeiros 64 bytes de **/etc/passwd**", o kernel poderia traduzir esta solicitação numa instrução de driver de dispositivo como "Recupere o bloco 3.348 do dispositivo 3". O driver subdividiria ainda mais esse comando em seqüências de padrões de bits a serem apresentadas aos registradores de controle do dispositivo.

O kernel é, em grande parte, escrito em C, com uma pitada de linguagem assembly para ajudá-lo a interfacear com funções específicas de hardware ou de chips que não são acessíveis através das diretivas de compilador normais.

Uma das vantagens do ambiente Linux é que a disponibilidade de código-fonte torna relativamente fácil você desenvolver os seus próprios drivers de dispositivos e módulos de kernel a partir da estaca zero. Nos primórdios do Linux, ter habilidades nesse campo era uma necessidade pois era difícil administrar sistemas Linux efetivamente sem ser capaz de "moldar" o sistema para um ambiente específico.

Hoje, o Linux é tão universal que os administradores de sistemas podem ser perfeitamente eficientes sem jamais ter suas mãos sujas com código kernel. Na realidade, pode-se argumentar que tais atividades são melhor tratadas pelas mãos de programadores e que o foco dos administradores deveria ser mais em relação às necessidades como um todo da comunidade de usuários. Administradores de sistemas podem ajustar o kernel ou adicionar módulos preexistentes conforme descrito neste capítulo, porém eles não precisam fazer um curso aprofundado de programação em C ou linguagem assembly 80x86 para sobreviverem.

O objetivo final é que você não confunda a administração de ambientes Linux modernos com a "economia doméstica" pouco explorada de alguns anos atrás.

www.snow666.blogspot.com
Snow666

12.1 Adaptação do Kernel

Os sistemas Linux vivem num mundo que poderia incluir potencialmente qualquer um das dezenas de milhares de dispositivos de hardware de computadores disponíveis hoje. O kernel tem de se adaptar a seja lá qual for o hardware presente na máquina na qual estiver sendo executada.

Um kernel pode aprender sobre o hardware do sistema de uma série de maneiras. As mais básicas são você fornecer ao kernel informações explícitas sobre o hardware que ele deve esperar encontrar (ou espera não encontrar, como pode ser o caso). Alguns kernels podem também tentar encontrar dispositivos por conta própria, seja no momento da inicialização ou dinamicamente assim que o sistema estiver no ar.

Na plataforma PC, em que o Linux é popular, o desafio de criar um inventário acurado do hardware do sistema é particularmente difícil (e, algumas vezes, impossível). O hardware de PC seguiu uma linha evolutiva não muito diferente da nossa na qual protozoários primitivos agora deram vida a tudo, de cães selvagens a abelhas assassinas. Essa diversidade é composta pelo fato de que fabricantes de PCs normalmente não nos oferecem muita informação sobre os sistemas que eles vendem, de modo que normalmente temos de "desmontar" nosso sistema e inspecionar visualmente as peças para responder a questões como "Que chipset minha placa de rede usa?".

A maior parte dos sistemas Linux modernos sobrevive sob uma dieta híbrida de componentes de kernel estáticos e dinâmicos, com a mistura entre os dois sendo ditada basicamente pelas limitações de hardware do PC. É provável que em algum ponto durante a sua carreira de administrador de sistemas você precisará dar uma mão na forma de criar uma configuração de kernel.

12.2 Por Que Configurar o Kernel?

Quando o sistema é instalado, ele vem com uma configuração genérica que é desenhada para executar a maioria dos aplicativos na maioria dos dispositivos de hardware. A configuração genérica inclui muitos drivers de dispositivos diferentes e pacotes opcionais e ela possui valores de parâmetros ajustáveis escolhidos para "emprego genérico". Examinando-se cuidadosamente esta configuração e ajustando-a para suas necessidades exatas, você pode ser capaz de otimizar o desempenho, a segurança e até mesmo a confiabilidade de seu sistema.

Os kernels de Linux modernos são melhores do que seus antecessores ao descarregarem drivers indesejados da memória, porém opções compiladas não como módulos sempre estarão ativas. Embora reconfigurar o kernel para fins de eficiência seja menos importante do que costumava ser, um bom caso ainda pode justificar fazê-lo.

Instruções para adicionar um novo driver estão na seção 12.5.

Uma outra razão para reconfigurar o kernel é a de adicionar suporte a novos tipos de dispositivos (por exemplo, adicionar novos drivers de dispositivos). O código do driver não pode simplesmente ser enfiado no kernel como um monte de massinha de modelar; ele tem de ser integrado às tabelas e estruturas de dados do kernel. Em alguns sistemas, esse procedimento pode exigir que você retorne aos arquivos de configuração para o kernel e adicione o novo dispositivo, reconstruindo o kernel da estaca zero. Em outros sistemas, talvez você só precise executar um programa desenvolvido para fazer automaticamente essas mudanças de configuração.

O kernel não é difícil de ser configurado; é apenas difícil corrigir uma vez que o danifiquemos.

12.3 Métodos de Configuração

Quatro métodos básicos podem ser usados para configurar o kernel do Linux. Provavelmente você terá a oportunidade de, eventualmente, tentar todos eles. Os métodos são:

- Modificar parâmetros de configuração de kernel ajustáveis (dinâmicos).
- Carregar novos drivers e módulos no kernel existente durante a execução.
- Criar um kernel da estaca zero (realmente, isto significa compilá-lo de arquivos-fonte, possivelmente com modificações e acréscimos).
- Fornecer diretivas operacionais no momento do boot através do loader de kernel, LILO ou GRUB. Consulte a página 19 para obter maiores informações sobre esses sistemas.

Cada um desses métodos é aplicável em situações ligeiramente diferentes. Modificar parâmetros ajustáveis é a maneira mais fácil e mais comum, ao passo que criar um kernel de arquivos-fonte é o jeito mais difícil e normalmente menos exigido. Felizmente, todas essas abordagens acabam sendo incorporadas com um pouco de prática.

12.4 Ajustando um Kernel Linux

Muitos módulos e drivers no kernel foram desenvolvidos com o conhecimento de que uma condição não atende a tudo. Para aumentar a flexibilidade, ganchos especiais permitem que parâmetros como o tamanho de uma tabela in-

Snow666

terna ou o comportamento do kernel numa determinada circunstância sejam ajustados pelo administrador de sistemas com o sistema em execução. Esses ganchos são acessíveis por meio de uma abrangente interface kernel-usuários representada por arquivos no sistema de arquivos **/proc**. Em muitos casos, um grande aplicativo em nível de usuário (especialmente um aplicativo de infra-estrutura como um banco de dados) exigirá o ajuste de parâmetros para atender às suas necessidades específicas.

Arquivos especiais em **/proc/sys** nos permitem visualizar e configurar opções de kernel em tempo de execução. Esses arquivos simulam arquivos Linux padrão, porém eles são realmente portas dos fundos para o kernel. Se um desses arquivos tiver um valor que você pretenda alterar, poderá tentar gravar nele. Infelizmente, nem todos os arquivos podem ser gravados (independentemente de suas permissões aparentes) e nenhuma documentação informa-lhe sobre quais podem ou não ser gravados.

Por exemplo, para alterar o número máximo de arquivos abertos que um processo pode ter, tente algo como

```
# echo 32768 > /proc/sys/fs/file-max
```

Assim que se acostumar com essa interface não ortodoxa, você achará bem útil, especialmente para modificar opções de configuração. Uma palavra de alerta, porém: mudanças não são lembradas após reinicializações. Se quiser fazer alterações permanentes, você deve usar o arquivo **/etc/sysctl.conf** documentado a seguir. A Tabela 12.1 apresenta algumas das opções úteis.

Tabela 12.1 Arquivos em /proc/sys para alguns exemplos de parâmetros de kernel ajustáveis.

Dir[a]	Arquivo	Padrão	Função
F	**file-max**	4096	Configura o nº máximo de arquivos abertos por processo
F	**inode-max**	16384	Configura o nº máximo de inodes abertos por processo
N	**ip_forward**	0	Permite encaminhamento IP quando configurado em 1
N	**icmp_echo_ignore_all**	0	Ignora pings ICMP quando configurado em 1
N	**icmp_echo_ignore_broadcasts**	0	Ignora pings de difusão ampla quando configurado em 1
K	**shmmax**	32M	Configura o tamanho máximo de memória compartilhada
K	**ctrl-alt-del**	0	Reinicializa na seqüência Ctrl-Alt-Del?
C	**autoeject**	0	Ejeção automática do CD-ROM na desmontagem?

a. F=**/proc/sys/fs**, N=**/proc/sys/net/ipv4**, K=**/proc/sys/kernel**, C=**/proc/sys/dev/cdrom**

Consulte a seção 13.10 para aprender sobre parâmetros de rede adicionais que são ajustáveis.

Uma maneira menos provisória para modificar esses mesmos parâmetros pode ser encontrada na maioria dos sistemas na forma do comando **sysctl**. **sysctl** pode configurar variáveis individuais seja pela linha de comando seja lendo uma lista da pares *variável=valor* de um arquivo. Como padrão, o arquivo **/etc/sysctl.conf** é lido no momento do boot e seu conteúdo usado para configurar os valores de parâmetros iniciais (personalizados).

Por exemplo, o comando

```
# sysctl net.ipv4.ip_forward=0
```

desativa o encaminhamento de IPs. Note que formamos os nomes das variáveis usadas por **sysctl** substituindo os /'s na estrutura de diretórios **/proc/sys** por pontos.

12.5 Adicionando Drivers de Dispositivos

Um driver de dispositivo é um programa que gerencia a interação do sistema com um dispositivo de hardware. O driver faz a tradução entre os comandos de hardware compreendidos pelo dispositivo e a interface de programação estilizada usada pelo kernel. A existência da camada de driver ajuda a manter o Linux relativamente independente de dispositivos.

Drivers de dispositivos fazem parte do kernel; eles não são processos de usuários. Porém, um driver pode ser acessado tanto dentro do kernel quanto do espaço de usuário. O acesso em nível de usuário para dispositivos é normalmente fornecido através de arquivos de dispositivos especiais que residem no diretório **/dev**. O kernel transforma operações nesses arquivos em chamadas para o código do driver.

A plataforma PC tem sido uma fonte de caos no mundo de administrador de sistemas. Um conjunto desconcertante de hardware e de "padrões" com níveis variáveis de suporte por parte de sistemas operacionais se encontram à disposição. Veja:

www.snow666.blogspot.com
Snow666

- Mais de 30 chipsets SCSI diferentes são suportados pelo Linux e cada um deles é empacotado e vendido pelo menos pelo dobro deste número de fornecedores.
- Mais de 200 interfaces de rede diferentes se encontram lá, cada uma delas sendo comercializada por vários fornecedores diferentes sob nomes diversos.
- Tipos de hardware mais novos, melhores e mais baratos estão em desenvolvimento a todo instante. Cada um deles requer um driver de dispositivo para poder funcionar com o sistema Linux de sua preferência.

Com esse ritmo alucinante no qual novos dispositivos de hardware estão sendo desenvolvidos, é praticamente impossível manter atualizadas as principais distribuições de sistemas operacionais em relação ao hardware mais recente. Não é nada incomum termos de adicionar um driver de dispositivo ao nosso kernel para suportar um novo dispositivo de hardware.

Somente drivers de dispositivos para uso com o Linux (e normalmente, uma versão específica do kernel Linux) pode ser instalada com sucesso num sistema Linux. Drivers para outros sistemas operacionais (como o Windows) não funcionarão, de modo que na hora de adquirir novo hardware terá de ter isto em mente. Além disso, os dispositivos variam em termos de nível de compatibilidade e de funcionalidade quando usados com o Linux; ou seja, preste atenção aos resultados obtidos por outras instalações com qualquer hardware que esteja considerando comprar.

Os fornecedores estão se tornando mais cientes do mercado Linux e ocasionalmente eles até fornecem drivers Linux. Você pode ter sorte e descobrir que o seu fornecedor irá lhe colocar à disposição tanto drivers como suas respectivas instruções para instalação. Mais provavelmente, você somente encontrará o driver necessário em alguma página Web não comentada. Em qualquer caso, esta seção lhe mostra o que está realmente acontecendo quando adicionamos um driver de dispositivo.

Números de dispositivos

Muitos dispositivos possuem um arquivo correspondente em **/dev**, sendo as exceções notáveis em sistemas operacionais modernos os dispositivos de rede. Por força de serem arquivos de dispositivos, os arquivos em **/dev** possuem um número de dispositivo principal e um secundário associado a eles. O kernel usa esses números para associar referências a um arquivo de dispositivo ao driver correspondente.

O número de dispositivo principal identifica o driver com o qual o arquivo está associado (em outras palavras, o tipo de dispositivo). O número de dispositivo secundário identifica qual instância em particular de um dado tipo de dispositivo deve ser tratada. O número de dispositivo secundário é algumas vezes chamado de número de unidade.

Podemos ver os números de dispositivo principal e secundário de um arquivo de dispositivo pelo comando **ls -l**:

```
% ls -l /dev/sda
brw-rw---- 1 root     disk     8,   0 Mar 3   1999 /dev/sda
```

Esse exemplo ilustra o primeiro disco SCSI num sistema Linux. Ele possui um número principal 8 e um secundário igual a 0.

O número de dispositivo secundário é algumas vezes usado pelo driver para selecionar a característica particular de um dispositivo. Por exemplo, uma única unidade de fita pode ter vários arquivos em **/dev** representando-a em várias configurações de densidade de gravação e características de rebobinamento. Em suma, o driver é livre para interpretar o número de dispositivo secundário da maneira que lhe aprouver. Veja a página de manual para o driver para determinar que convenção ele está adotando.

Há, na verdade, dois tipos de arquivos de dispositivos: arquivos de dispositivos de blocos e arquivos de dispositivos de caracteres. Um dispositivo de blocos é lido ou gravado um bloco por vez (um grupo de bytes, normalmente um múltiplo de 512); um dispositivo de caracteres pode ser lido ou gravado um byte por vez. Alguns dispositivos suportam acesso tanto através de arquivos de dispositivos de caracteres quanto de blocos, embora isto seja extremamente raro no Linux.

Algumas vezes é conveniente implementar uma abstração como um driver de dispositivo mesmo quando ele não controla efetivamente nenhum dispositivo. Tais dispositivos fantasmas são conhecidos como pseudodispositivos. Por exemplo, um usuário que faz o login numa rede recebe um PTY (pseudo-TTY) que se parece e se comporta como uma porta serial sob a perspectiva do software de alto nível. Esse truque permite que programas escritos nos dias em que todo o mundo usava uma TTY continuem a funcionar no mundo atual das janelas e das redes.

Quando um programa efetua uma operação sobre um dispositivo de arquivo, o kernel captura automaticamente a referência, procura o nome de função apropriado numa tabela e transfere o controle para ela. Para executar uma

Snow666

operação incomum que não tenha um análogo direto no modelo de sistema de arquivos (por exemplo, ejetar um disquete), um programa pode usar a chamada de sistema **ioctl** para passar uma mensagem diretamente do espaço de usuário para o driver.

12.6 ADICIONANDO UM DRIVER DE DISPOSITIVO LINUX

Em sistemas Linux, os drivers de dispositivo são distribuídos tipicamente em uma das três formas:

- Uma correção (patch) em oposição a uma versão de kernel específica
- Um módulo carregável
- Um script de instalação que aplica correções apropriadas

A mais comum delas é a correção em oposição a uma versão de kernel específica. Essas correções podem, na maioria dos casos, serem aplicadas com o seguinte procedimento:[1]

```
# cd /usr/src/linux ; patch -p1 < patch_file
```

Diferenças em relação a uma versão secundária diferente do kernel podem falhar, porém o driver ainda deve funcionar. Aqui, cobrimos como adicionar manualmente um driver de rede "ardiloso" ao kernel. Trata-se de um processo muito complicado e entediante, especialmente quando comparado a outros sistemas operacionais que já vimos.

Por convenção, o fonte do kernel Linux reside em **/usr/src/linux**. Dentro do subdiretório **drivers**, você precisará localizar o subdiretório correspondente ao tipo de dispositivo que possui. Uma listagem de diretórios de drivers se parece com o seguinte:

```
% ls -F /usr/src/linux/drivers
Makefile cdrom/    i2o/        nubus/      sbus/     telephony/
acorn/   char/     isdn/       parport/    scsi/     usb/
ap1000/  dio/      macintosh/  pci/        sgi/      video/
atm/     fc4/      misc/       pcmcia/     sound/    zorro/
block/   i2c/      net/        pnp/        tc/
```

Os diretórios mais comuns aos quais os drivers são adicionados são **block**, **char**, **net**, **usb**, **sound** e **scsi**. Esses diretórios contêm drivers para dispositivos de blocos (como as unidades de disco IDE), dispositivos de caracteres (como as portas seriais), dispositivos de rede, dispositivos USB, placas de som e placas SCSI, respectivamente. Alguns dos outros diretórios contêm drivers para os barramentos em si (por exemplo, pci, nubus e zorro); é pouco provável que você precisará adicionar drivers a esses diretórios. Alguns diretórios contêm drivers de plataforma específica como **macintosh**, **acorn** e **ap1000**. Outros diretórios contêm dispositivos especiais como **atm**, **isdn** e **telephony**.

Já que nosso exemplo de dispositivo é um relacionado à rede, adicionaremos o driver ao diretório **drivers/net**. Precisaremos modificar os seguintes arquivos:

- **drivers/net/Makefile**, de modo que nosso driver seja compilado.
- **drivers/net/Config.in**, de modo que nosso dispositivo apareça nas opções de configuração.
- **drivers/net/Space.c**, de modo que o dispositivo seja testado na inicialização.

Após colocarmos os arquivos **.c** e **.h** para o driver em **drivers/net**, adicionaremos o driver a **drivers/net/Makefile**. As linhas que adicionaremos (próximo ao final do arquivo) são apresentadas a seguir:

```
ifeq ($(CONFIG_SNARF),y)
      L_OBJS += snarf.o
else
      ifeq ($(CONFIG_SNARF),m)
      M_OBJS += snarf.o
      endif
endif
```

Essa configuração acrescenta o driver snarf parta que ele seja tanto configurado como um módulo quanto embutido no kernel.

1. Obviamente, o pacote de kernel original tem de ser instalado antes de modificarmos a árvore do kernel.

www.snow666.blogspot.com
Snow666

Após adicionarmos o dispositivo ao **Makefile**, temos de ter certeza de que podemos configurar o dispositivo ao configurarmos o kernel. Todos os dispositivos de rede precisam ser listados no arquivo **drivers/net/Config.in**. Para adicionarmos o dispositivo de forma que ele possa ser construído como um módulo ou, então, como parte do kernel (consistente com o que solicitamos no **Makefile**), adicionamos a linha a seguir:

```
tristate 'Suporte ao dispositivo snarf' CONFIG_SNARF
```

A palavra-chave `tristate` significa que podemos criar o dispositivo como um módulo. Se o dispositivo não puder ser construído como um módulo, use a palavra-chave `bool` em vez de `tristate`. O próximo identificador é a string a ser exibida na tela de configuração. Ele pode ser qualquer texto arbitrário, porém deve identificar o dispositivo que está sendo configurado. O identificador final é a macro de configuração. Esse identificador precisa ser o mesmo que aquele testado com a instrução `ifeq` no **Makefile**.

O último arquivo que precisamos editar para adicionar nosso dispositivo ao sistema é **drivers/net/Space.c**. **Space.c** contém referências às rotinas de sondagem para o driver de dispositivo e também controla a ordem da sondagem de dispositivo. Aqui, teremos de editar o arquivo em dois lugares distintos. Primeiramente acrescentaremos uma referência à função de sondagem, depois adicionaremos o dispositivo à lista de dispositivos a serem sondados.

No início do arquivo **Space.c** temos um bocado de referências a outras funções de sondagem. Acrescentaremos a seguinte linha a essa lista:

```
extern int snarf_probe(struct device *);
```

Depois disso, para acrescentarmos o dispositivo à lista de sondagem em si, precisamos determinar à qual lista devemos adicioná-lo. Uma lista de sondagem distinta é mantida para cada tipo de barramento (PCI, EISA, SBus, MCA, ISA, porta paralela etc.). O dispositivo snarf é um dispositivo PCI, de modo que o adicionaremos à lista chamada `pci_probes`. A linha que diz

```
struct devprobe pci_probes[] __initdata = {
```

é seguida por uma lista ordenada de dispositivos. Os dispositivos logo no início da lista são os primeiros a serem sondados. A ordem de sondagem normalmente não interessa para os dispositivos PCI, porém alguns dispositivos são sensíveis. Apenas para certificar-nos de que o dispositivo snarf será detectado, o acrescentaremos ao início da lista:

```
struct devprobe pci_probes[] __initdata = {
#ifdef CONFIG_SNARF
   snarf_probe, 0},
endif
```

O dispositivo foi agora adicionado ao kernel Linux. Quando depois configurarmos o Kernel, o dispositivo deverá aparecer como uma opção de configuração sob "dispositivos de rede".

12.7 ARQUIVOS DE DISPOSITIVOS

Por convenção, os arquivos de dispositivos são mantidos no diretório **/dev**. Sistemas grandes, especialmente aqueles com rede e pseudoterminais, podem suportar centenas de dispositivos.

Os arquivos de dispositivos são criados pelo comando **mknod**, que possui a seguinte sintaxe

```
mknod filename type major minor
```

As atribuições oficiais de números de dispositivos no Linux podem ser encontradas em www.lanana.org.

onde *filename* é o arquivo de dispositivos a ser criado, *type* é **c** para um dispositivo de caracteres ou **b** para um dispositivo de blocos e *major* e *minor* são os números de dispositivo principal e secundário. Se estiver criando um arquivo de dispositivos que se refere a um driver que já esteja presente em seu kernel, verifique a página de manual referente ao driver para localizar os números principal e secundário apropriados.

O Red Hat e Debian fornecem um script chamado **MAKEDEV** (em **/dev**) para fornecer automaticamente valores-padrão para **mknod**. Estude o script para descobrir os argumentos necessários para seu dispositivo. Por exemplo, para criarmos entradas PTY, usaríamos os seguintes comandos:

```
# cd /dev
# ./MAKEDEVICE pty
```

Snow666

Convenções de nomes para dispositivos

As convenções para atribuição de nomes de dispositivos são um tanto aleatórias. Elas normalmente são remanescentes da maneira como era feita no UNIX num DEC PDP-11, por mais arcaico que isso possa parecer nos dias de hoje.

Consulte o Capítulo 7 para obter maiores informações sobre dispositivos seriais.

Os arquivos de dispositivos seriais são chamados **ttyS** seguido por um número que identifica a interface específica à qual a porta está conectada. TTYs são algumas vezes representados por mais de um arquivo de dispositivo; os arquivos extras normalmente fornecem acesso a métodos de controle de fluxo alternativos ou a protocolos de bloqueio.

Os nomes de dispositivos de fita normalmente incluem não somente uma referência à unidade em si, como também uma indicação de o dispositivo rebobinar após cada operação com fita e a densidade na qual ele lê e grava.

Os dispositivos de disco rígido IDE recebem nomes **/dev/hd**LP, onde L é uma letra que identifica a unidade (com **a** sendo o mestre na primeira interface IDE, **b** sendo o escravo nesta mesma interface, **c** sendo o mestre na segunda interface IDE e assim por diante) e P é o número de partição (começando com 1). Por exemplo, a primeira partição do primeiro disco IDE é tipicamente **/dev/hda1**. Os discos SCSI recebem nomes de forma similar, porém com o prefixo **/dev/sd** em vez de **/dev/hd**. Podemos eliminar o número de partição em ambos os tipos de dispositivos para poder acessar o disco inteiro (por exemplo, **/dev/hda**).

As unidades de CD-ROM SCSI são designadas pelos arquivos **/dev/scd**N, onde N é um número que distingue várias unidades de CD-ROM. As unidades de CD-ROM modernas IDE (ATAPI) são designadas da mesma maneira que os discos rígidos IDE (por exemplo, **/dev/hdc**).

12.8 Módulos de Kernel Carregáveis

O suporte a módulos de kernel carregáveis (LKM - Loadable Kernel Module) permite que um driver de dispositivo — ou qualquer um outro serviço de kernel — seja linkado e removido do kernel enquanto ele está em execução. Esse recurso torna muito mais fácil a instalação de drivers, já que o binário de kernel não precisa ser modificado. Ele também possibilita que o kernel seja menor, pois os drivers não são carregados a menos que necessário.

Módulos carregáveis são implementados por um ou mais "ganchos" documentados no kernel aos quais drivers de dispositivo adicionais podem se "pendurar". Um comando em nível de usuário se comunica com o kernel e o informa para carregar novos módulos na memória. Há normalmente um comando que descarrega drivers também.

Embora os módulos carregáveis sejam convenientes, eles não são inteiramente seguros. A qualquer momento que carregamos ou descarregamos um módulo, corremos o risco de provocar um pânico de kernel. Não recomendamos ficar carregando e descarregando um módulo não testado quando não estiver a fim de provocar uma pane na máquina.

Sob o Linux, quase tudo pode ser construído como um módulo de kernel carregável. As exceções ficam por conta do tipo de sistema de arquivos raiz, o dispositivo no qual o sistema de arquivos raiz reside e o driver de mouse PS/2.

Os módulos de kernel carregáveis são convencionalmente armazenados em **/lib/modules/***version*, onde *version* é a versão de seu kernel Linux retornada por **uname -r**. Podemos inspecionar os módulos atualmente carregados por meio do comando **lsmod**:

```
# lsmod
Module          Size     Used by
PPP             21452    0
slhc             4236    0 [PPP]
ds               6344    1
i82365          26648    1
pcmcia_core     37024    0 [ds i82365]
```

Essa máquina possui os módulos de controlador PCMCIA, o driver PPP e os módulos de compactação de cabeçalhos PPP carregados.

Os LKMs Linux podem ser carregados manualmente no kernel por meio de **insmod**. Poderíamos, por exemplo, inserir manualmente nosso módulo snarf com o comando

```
# insmod /path/to/snarf.o
```

Parâmetros também podem ser passados para módulos de kernel carregáveis; por exemplo,

```
# insmod /path/to/snarf.o io=0xXXX irq=X
```

Uma vez que um módulo de kernel carregável tenha sido inserido manualmente no kernel, ele somente será removido se solicitarmos explicitamente sua eliminação ou se o sistema for reinicializado . Poderíamos usar **rmmod snarf** para remover nosso módulo snarf.

Snow666

Podemos usar **rmmod** a qualquer momento, porém ele funciona somente se o número de referências atuais ao módulo (listado na coluna `Used` da saída gerada pelo comando **lsmod**) for 0.

Também podemos carregar LKMs Linux de forma semi-automática com **modprobe**, um envoltório para **insmod** que compreende as dependências, opções e procedimentos de instalação/remoção. **modprobe** usa o arquivo **/etc/modules.conf** para descobrir como tratar cada um dos módulos.

Podemos gerar um arquivo **/etc/modules.conf** correspondente a todos os nossos módulos instalados no momento executando **modprobe -c**. Esse comando gera um arquivo longo parecido com o seguinte:

```
# Este arquivo foi gerado por: modprobe -c (2.1.121)
path[pcmcia]=/lib/modules/preferred
path[pcmcia]=/lib/modules/default
path[pcmcia]=/lib/modules/2.3.39
path[misc]=/lib/modules/2.3.39
...
# Nomes Alternativos
alias block-major-1 rd
alias block-major-2 floppy
...
alias char-major-4 serial
alias char-major-5 serial
alias char-major-6 lp
...
alias dos msdos
alias plip0 plip
alias ppp0 ppp
options ne io=x0340 irq=9
```

As declarações `path` informam onde determinado módulo pode ser encontrado. Podemos modificar ou adicionar entradas desse tipo caso queiramos manter nossos módulos numa localização não padrão.

A declaração `alias` fornece uma associação entre números principais de dispositivos de blocos, números principais de dispositivos de caracteres, sistemas de arquivos, dispositivos de redes e protocolos de rede e os nomes dos módulos correspondentes.

As linhas `options` não são geradas dinamicamente. Elas especificam opções que deveriam ser passadas para um módulo quando ele é carregada. Poderíamos, por exemplo, usar a linha a seguir para informar ao módulo snarf seu endereço de I/O e vetor de interrupção apropriados:[2]

```
options snarf io=0xXXX irq=X
```

modprobe também compreende as instruções `pre-install`, `post-install`, `pre-remove`, `post-remove`, `install` e `remove`. Essas instruções permitem que comandos sejam executados quando um módulo específico é inserido ou removido do kernel em execução. Elas assumem a seguinte forma

```
pre-install module command ...
install module command ...
post-install module command ...
pre-remove module command ...
remove module command ...
post-remove module command ...
```

e são executadas antes da inserção, simultaneamente com a inserção (se possível), após a inserção, antes da remoção, durante a remoção (se possível) e após a remoção.

Os sistemas Debian executam **/etc/cron.daily/modutils** uma vez por dia para despachar módulos que abusaram de sua receptividade no kernel.

2. Se estiver usando hardware PC, pode ser um desafio criar uma configuração na qual os vetores de solicitação de interrupção (IRQs) e portas I/O não sejam coincidentes. Podemos ver as atribuições atuais em um sistema examinando, respectivamente, o conteúdo de **/proc/interrupts** e **/proc/ioports**.

www.snow666.blogspot.com
Snow666

12.9 Construindo um Kernel Linux

Pelo fato de o Linux estar evoluindo tão rapidamente, é muito mais provável que você irá se deparar com a necessidade de construir um kernel Linux do que se estivesse executando um sistema operacional de grande porte. As correções de kernel, os drivers de dispositivos e nova funcionalidade estão constantemente sendo disponibilizados. Isso realmente é algo meio abençoado. Por um lado, é conveniente sempre ter de suportar o "mais recente e o melhor", por outro lado pode tomar muito tempo manter-se atualizado com tamanho fluxo de material novo. Mas após você ser bem-sucedido na construção de um kernel, se sentirá fortalecido e pronto para fazê-lo de novo.

É menos provável que você precisará construir um kernel por conta própria caso esteja rodando uma versão "estável". O Linux adotou um esquema de numeração de versões no qual a segunda parte do número da versão indica se o kernel é estável (números pares) ou ainda está em desenvolvimento (números ímpares). Por exemplo, o kernel versão 2.4.3 seria um kernel "estável", ao passo que 2.5.3 seria um kernel "de desenvolvimento". Agora você já sabe.

A configuração do kernel Linux trilhou um longo caminho, mas ainda parece muito primitiva comparada com o procedimento usado em alguns outros sistemas. O processo gira em torno do arquivo **.config** na raiz do diretório dos fontes do kernel (normalmente **/usr/src/linux**).[3] Todas as informações de configuração de kernel são especificadas nesse arquivo, porém seu formato é um tanto criptográfico. Use o guia de decodificação em **Documentation/Configure.help** para descobrir o que as diversas opções significam.

Para poupar os amigos de ter de editar o arquivo **.config** diretamente, o Linux possui várias diretivas do **make** que nos permitem configurar o arquivo kernel com diferentes interfaces. Se estiver rodando o X Windows, a melhor interface de configuração será fornecida por **make xconfig**. Esse comando aciona uma tela de configuração gráfica na qual podemos selecionar os dispositivos a serem adicionados ao kernel (ou compilar como módulos carregáveis).

Se você não estiver executando o X Windows, poderá usar uma alternativa baseada em **curses**[4] chamada por meio de **make menuconfig**. Finalmente há o velho estilo **make config**, que lhe apresenta um prompt para responder a cada opção de configuração disponível sem lhe permitir retornar e mudar de idéia.

Recomendamos **make xconfig** caso esteja rodando o X Windows e **make menuconfig** caso não esteja. Evite **make config**.

Essas ferramentas são bem objetivas no que diz respeito às opções que podemos ativar, porém, infelizmente, elas são difíceis de usar caso queiramos manter várias versões do kernel para várias arquiteturas ou configurações de hardware.

As diversas interfaces de configuração descritas acima geram um arquivo **.config** que se parece com o seguinte:

```
# make config gerado automaticamente: não edite
#
# Opções de nível de maturidade de código

CONFIG_EXPERIMENTAL=y
#
# Tipo de processador e recursos
#
# CONFIG_M386 não está configurado
# CONFIG_M486 não está configurado
# CONFIG_M586 não está configurado
# CONFIG_M586TSC não está configurado
CONFIG_M686=y
CONFIG_X86_WP_WORKS_OK=y
CONFIG_X86_INVLPG=y
CONFIG_X86_BSWAP=y
CONFIG_X86_POPAD_OK=y
CONFIG_X86_TSC=y
CONFIG_X86_GOOD_APIC=y
...
```

Como pode ser visto, o conteúdo é um tanto enigmático e não fornece nenhuma descrição sobre o significado das tags `CONFIG`. Algumas vezes podemos descobrir o significado. Basicamente, cada linha `CONFIG` se refere a uma opção de configuração de kernel específica. O valor **y** compila a opção no kernel; **m** o habilita, porém como um módulo carregável.

---

3. Repetindo, é necessário instalarmos o pacote de kernel original antes de poder construir um kernel em um sistema; veja a página 547 para dicas sobre a instalação de pacotes.
4. **curses** é uma biblioteca da época em que se costumava criar GUIs baseadas em texto que rodavam numa janela de terminal.

Snow666

Algumas coisas podem ser configuradas como módulos e outras não. Você somente precisa saber quem é quem; não fica claro pelo arquivo **.config**. Também não há uma associação fácil das tags `CONFIG` com informações significativas. Porém, podemos normalmente extrair essas informações do arquivo **Config.in** localizado em cada diretório de driver. Os arquivos **Config.in** são difíceis e inconvenientes de serem rastreados, de modo que é melhor usarmos simplesmente **make xconfig** ou **make menuconfig**.

Assim que tiver um kernel funcionando e operacional, talvez você precise passar opções de configuração especiais a ele no momento da inicialização, como o dispositivo-raiz que ele deve usar ou uma instrução para sondar várias placas Ethernet. O boot loader (tipicamente LILO ou GRUB) repassa essas opções. Adicionamos opções de configuração estáticas ao arquivo **/etc/lilo.conf** ou a **/boot/grub/grub.conf**, dependendo de qual boot loader usamos. Veja o Capítulo 2 para maiores informações.

Se não for possível editarmos o arquivo de configuração do boot loader (talvez por termos danificado algo e a máquina não conseguir mais fazer o boot), podemos passar as opções manualmente. Por exemplo, no prompt de boot do LILO poderíamos digitar

```
LILO: linux root=/dev/hda1 ether=0,0,eth0 ether=0,0,eth1
```

para informar o LILO para carregar o kernel especificado pela tag "linux", para usar o dispositivo-raiz **/dev/hda1** e para sondar duas placas Ethernet.

Um exemplo similar usando GRUB seria parecido com o seguinte:

```
grub> kernel /vmlinuz root=/dev/hda1 ether=0,0,ether=0 ether=0,0,eth1
grub> boot
```

Construindo o binário do kernel Linux

Configurar um arquivo **.config** apropriado é a parte mais importante do processo de configuração do kernel Linux, mas você terá de passar por muitas outras etapas para transformar este arquivo num kernel finalizado.

Eis uma descrição do processo todo:

- **cd** para **/usr/src/linux**.
- Executar **make xconfig** ou **make menuconfig**.
- Executar **make dep**.
- Executar **make clean**.
- Executar **make bzImage**.
- Executar **make modules**.
- Executar **make modules_install**.
- Copiar **/usr/src/linux/arch/i386/boot/bzImage** para **/boot/vmlinuz**.
- Copiar **/usr/src/linux/arch/i386/boot/System.map** para **/boot/System.map**.
- Editar **/etc/lilo.conf** (LILO) ou **/boot/grub/grub.conf** (GRUB) para adicionar uma linha de configuração para o novo kernel.
- Se estiver usando LILO, execute **/sbin/lilo** para instalar o boot loader reconfigurado.

A etapa **make clean** nem sempre é estritamente necessária, porém em geral é uma boa idéia começar com um ambiente de construção limpo. Na prática, muitos problemas podem ser descobertos ao pularmos essa etapa.

O Debian fornece um utilitário **make-kpkg** que automatiza algumas das etapas acima e permite a organização de opções de compilação do kernel. Veja a página de manual do **make-kpkg** para maiores informações sobre sua utilização e configuração.

12.10 Não Conserte se Não Está Quebrado

Com as novas versões do kernel Linux entrando em cena a cada período de poucos meses e novos drivers e correções sendo lançados a cada dia, é fácil se tornar viciado em correções e atualizações. Afinal de contas, o que há de mais excitante do que informar à sua comunidade de usuário que você acabou de descobrir uma nova correção de kernel e que estará desativando o servidor de e-mails durante o período da tarde para instalá-la? Alguns administradores justificam suas existências dessa maneira; todo o mundo gosta de bancar o herói.

Um bom administrador de sistemas pesa cuidadosamente necessidades e riscos ao planejar atualizações e correções a serem aplicadas no kernel. Certamente, a nova versão pode ser a mais recente e a mais sensacional, porém

Snow666

ela é tão estável quanto a versão atual? Essa atualização ou correção poderia ser adiada e instalada com um outro grupo de correções no final do mês? É importante resistir à tentação de deixar o vamos "nos manter igual aos nossos camaradas" (neste caso, a comunidade hacker apaixonada por kernels) prevalecer sobre os interesses de sua comunidade de usuários.

Uma boa regra prática é atualizar ou aplicar correções somente quando os ganhos de produtividade que você espera obter (normalmente medidos em termos de confiabilidade e de desempenho) forem exceder o esforço e o tempo gastos para realizar a instalação. Caso esteja com problemas em quantificar o ganho específico, isto é um bom sinal de que a correção pode esperar mais alguns dias.

12.11 LEITURA RECOMENDADA

BECK, Michael, et al. *Linux Kernel Internals. 2. ed.* Reading, MA: Addison-Wesley. 1998.

Trata-se de um antigo livro sobre o kernel do Linux, porém ainda é o nosso favorito. Embora esteja um tanto desatualizado, ele dá uma boa explicação sobre o funcionamento interno do kernel.

12.12 EXERCÍCIOS

E12.1 Descreva o que o kernel faz. Explique as diferenças entre um kernel que usa módulos e um que não usa.

E12.2 Para cada método de configuração apresentado na seção 12.3, dê um exemplo prático de um parâmetro que poderia ser modificado.

★ **E12.3** Examine os valores de vários parâmetros da Tabela 12.1, utilizando tanto o método **/proc** quanto o método **sysctl**. Modifique dois dos valores com um método e depois os leia novamente com o outro método. Verifique se o comportamento do sistema mudou efetivamente em resposta aos ajustes que você aplicou. Emita um relatório escrito de seu experimento. (Requer acesso root.)

★ **E12.4** Num mercado de usados e raridades você fez um excelente negócio numa placa de laptop que lhe dá conectividade Ethernet através da porta paralela. Que etapas você precisaria realizar para fazer com que o Linux reconhecesse esta nova placa? Você deveria compilar suporte diretamente no kernel ou adicioná-lo na forma de um módulo? Por quê? (Pergunta extra: se você cobra R$ 80/hora de consultoria, estime o valor do trabalho necessário para fazer com que esta placa de rede baratinha funcione.)

★ **E12.5** Acaba de ser lançada uma nova versão do kernel Linux e você quer atualizar todas as máquinas no laboratório dos alunos (cerca de 50 máquinas, nem todas idênticas). Que questões você deveria considerar? Que procedimento você deveria seguir? Que problemas poderiam surgir e como você os trataria?

★★ **E12.6** No laboratório, configure um kernel com **xconfig** ou **menuconfig** e construa um binário de kernel. Instale e execute o novo sistema. Extraia a saída gerada por **dmesg** do kernel antigo e do novo e destaque as diferenças. (Requer acesso root.)

Sn❄w666

SEÇÃO 2

REDES DE COMPUTADORES

www.sn0w666.blogspot.co
Sn❄w666

Página em branco

Snow666

13 Redes TCP/IP

Seria difícil aumentar ainda mais a importância das redes para a computação moderna, embora isto pareça não impedir as pessoas de ficarem tentando. Em muitas instalações, os acessos à Web e ao e-mail agora são as principais atividades nas quais os computadores são usados. Na metade de 2001, estimava-se que a Internet tivesse cerca de 450 milhões de usuários e este número parece ainda estar crescendo rapidamente.[1] A manutenção de redes locais, conexões com a Internet, sites Web e software relacionado com redes é o arroz com feijão das tarefas da maior parte dos administradores de sistemas hoje em dia.

TCP/IP é a família de protocolos de rede usada no Linux/UNIX, MacOS, Windows e muitos outros sistemas operacionais. Ele também é a linguagem nativa da Internet. TCP quer dizer Transmission Control Protocol, ou seja, protocolo de controle de transmissão e IP significa Internet Protocol, ou seja, protocolo Internet.

Dispositivos que "falam" o protocolo TCP/IP são capazes de trocar dados (interoperarem) apesar das várias diferenças existentes entre eles. IP, o protocolo de remessa subjacente dessa família, é o burro de carga da Internet. TCP e UDP (User Datagram Protocol, protocolo de datagramas de usuário) são protocolos de transporte construídos sobre o IP para remeterem pacotes para aplicativos específicos.

O TCP é um protocolo orientado a conexões que facilita a conversação entre dois programas. Ele funciona de forma muito parecida com uma ligação telefônica: as palavras que você diz chegam à pessoa que você ligou e vice-versa. A conexão persiste mesmo quando nenhuma das partes está falando. O TCP fornece entrega confiável, controle de fluxo e controle de congestionamento.

O UDP é um serviço orientado a pacotes. Ele é análogo a enviar uma carta pelo correio. Não oferece uma comunicação nos dois sentidos, não possui qualquer forma de controle de congestionamento e não garante que os pacotes serão entregues na mesma ordem em que foram enviados.

O TCP é um protocolo educado que força os usuários que competem pelos mesmos recursos a compartilhar largura de banda e geralmente se comporta de modo que seja adequado para a produtividade da rede como um todo. Por outro lado, o UDP despeja pacotes o mais rápido que ele puder.

À medida que a Internet se torna cada vez mais popular e mais congestionada, precisamos que a maior parte do tráfego seja TCP para evitar congestionamento e compartilhar efetivamente a largura de banda disponível. Medidas de utilização de protocolos feitas ao longo dos últimos anos demonstraram que o tráfego UDP cresceu de 5% do total dos bytes transmitidos no período 1997-1998 para cerca de 7% do total transmitido no período 1999-2000. Em 2001 o TCP estava em ascensão novamente, com o UDP caindo para cerca de 3% a 4% do total de bytes e cerca de 12%

1. Vá ao endereço www.netsizer.com para ter uma idéia da estimativa atual.

Snow666

dos pacotes. Aplicativos como jogos, música, voz e vídeo estão fazendo notar suas presenças, porém são completamente sobrepujados pela Web e por programas como Gnutella que são verdadeiros comilões de largura de banda, mas que usam TCP em vez de UDP para transporte.

Este capítulo introduz os protocolos TCP/IP nos contextos técnico e político da Internet. Infelizmente, mesmo os conceitos básicos de rede são muito extensos para serem vistos num único capítulo. Outros capítulos relacionados a redes contidos neste livro são o Capítulo 14, *Roteamento*, Capítulo 20, *Gerenciamento de Redes e Depuração*, Capítulo 16, *DNS* e Capítulo 21, *Segurança*.

As próximas seções abrangem material básico sobre os protocolos e política da Internet e são bem contumazes e introdutórios. Pule para a seção 13.4 se quiser ir diretamente às entranhas do IP ou para a a seção 13.8 para ir diretamente a informações de configuração específicas das distribuições.

13.1 O TCP/IP e a Internet

O TCP/IP e a Internet compartilham uma história que retrocede a várias décadas. O sucesso técnico da Internet se deve em grande parte ao design elegante e flexível do TCP/IP e ao fato de que ele é uma família de protocolos aberta e não proprietária. Por outro lado, a alavancagem fornecida pela Internet ajudou o TCP/IP prevalecer sobre várias famílias de protocolo concorrentes que foram favorecidas num momento ou em outro por razões políticas e comerciais.

Um breve histórico

Contrário à crença popular, a Internet não é um produto da Microsoft que debutou em 1995, nem é a criação de um vice-presidente norte-americano. O progenitor da Internet moderna foi uma rede chamada ARPANET estabelecida em 1969 pela DARPA (Defense Advanced Research Project Agency), o braço de P&D do Departamento de Defesa dos Estados Unidos. A ARPANET tornou-se eventualmente o backbone NSFNET que conectava instalações com supercomputadores e redes regionais.

No final dos anos 80, a rede não era mais um projeto de pesquisa e chegara a hora da National Science Foundation pular fora do negócio das redes. A transição para a Internet comercial foi feita ao longo de um período de vários anos; a NSFNET foi desativada em abril de 1994. O backbone da Internet atual é um conjunto de redes privadas pertencentes a ISPs (Internet Service Providers, provedores de serviço Internet) que se interconectam em vários pontos denominados peer.

Em meados dos anos 80 a Internet era formada, basicamente, por instalações ARPANET originais e um punhado de universidades com computadores VAX da Digital rodando UNIX Berkley em redes Ethernet a 10 Mb/s e conectadas por linhas telefônicas digitais de 56 Kb alugadas. A cada mês de setembro, quando os estudantes retornavam às aulas, a Internet sofria o que acabou se tornando conhecido como colapso de congestionamento. Van Jacobson, então pesquisador do Grupo de Pesquisa em Redes do Lawrence Berkeley Labs, analisaria o comportamento dos protocolos sobre condições de carga e os corrigiria. Os algoritmos que hoje conhecemos como *partida lenta*, *impedimento de congestionamento*, *retransmissão rápida* e *recuperação rápida* surgiram desse contexto.

A lei de Moore (a regra prática de que a velocidade dos equipamentos dobram a cada 18 meses) e a pressão do mercado aceleraram em muito o desenvolvimento da Internet. Desde o final dos anos 80 quando os algoritmos TCP atuais foram estabilizados, a velocidade das interfaces de rede foi multiplicada por 1.000 (de 6% de eficiência das primeiras redes Ethernet com velocidade de 10 Mb/s para aproximadamente 90% de eficiência em redes Ethernet com velocidades na casa dos gigabits), a velocidade dos circuitos alugados multiplicou-se por 12.000 e o número total de hosts foi multiplicado por um fator de 50.000.

Em 1996, Bob Metcalfe, o inventor do Ethernet, previu que a Internet entraria em colapso, pois não seria instalada infra-estrutura em tempo hábil para acompanhar a demanda dos clientes, resultando em longas interrupções e uma rede que seria instável. Felizmente, a corrida ao ouro da Internet parece ter nos deixado com um excesso de capacidade e não uma falta de capacidade, pelo menos num futuro próximo.

Qualquer um que tivesse projetado um sistema de software e visse que ele havia se tornado obsoleto em relação à próxima geração de hardware ou à próxima versão de um sistema operacional sabe quão maravilhoso é o fato de nossa Internet ainda estar viva e resistindo, usando praticamente a mesma família de protocolos TCP/IP que foi elaborada há 25 anos para uma Internet muito diferente da atual. Tiramos o chapéu para Bob Kahn, Vint Cerf, Jon Postel, Van Jacobson e todas as demais pessoas que tornaram isso possível.

Como a Internet é gerenciada atualmente

O desenvolvimento da Internet sempre foi um empreendimento aberto e cooperativo. Agora que ela é uma força comercial mobilizadora na economia mundial, vários setores estão com medo e têm a impressão de que a Internet este-

Snow666

ja nas mãos de um punhado de geeks e com talvez uma pequena orientação por parte do governo americano. Agrade ou não, um controle sobre a Internet está se aproximando.

Várias organizações estão envolvidas:

- ICANN (Internet Corporation for Assigned Names and Numbers): se alguém pode ser visto como o administrador da Internet, este grupo é o tal (www.icann.org).
- ISOC (Internet Society): a ISOC é uma organização membro que representa a comunidade de usuários Internet (www.isoc.org).
- IETF (Internet Engineering Task Force): este grupo supervisiona o desenvolvimento e a padronização dos aspectos técnicos da Internet. Ele é o fórum aberto no qual qualquer um pode participar (www.ietf.org).

Desses grupos, a ICANN tem o trabalho mais árduo: estabelecer-se como a autoridade a cargo da Internet, desfazendo os erros do passado e antevendo o futuro.

Documentação e padrões de rede

As atividades técnicas da comunidade Internet são sintetizadas em documentos conhecidos como RFCs; uma RFC (Request for Comments) é uma solicitação de comentários. Os padrões de protocolos, as alterações propostas e os boletins informativos acabam tornando-se RFCs. As RFCs começam suas vidas como Internet Drafts e, após vários e-mails terem circulado de lá para cá e reuniões do IETF, elas acabam tendo um de dois destinos: ou morrem ou são promovidas para a categoria de RFC. Qualquer um que tenha algum comentário a respeito de um draft ou RFC proposto é incentivado a participar. Além de padronizar os protocolos Internet, o mecanismo RFC algumas vezes simplesmente documenta ou explica aspectos da prática existente.

As RFCs são numeradas seqüencialmente; atualmente, há cerca de 3.000 RFCs. Elas também possuem um título descritivo (por exemplo, *Alghoritms for Syncronizing Network Clocks*), porém para evitar ambigüidade elas são normalmente citadas pelo seu número. Uma vez distribuídas, o conteúdo de uma RFC jamais muda. As atualizações são distribuídas como novas RFCs com seus próprios números de referência. Por convenção, RFCs atualizadas contêm todo o material que permanece relevante, de forma que RFCs novas substituem completamente as antigas, pelo menos teoricamente.

O processo pelo qual as RFCs são publicadas é documentado em si na RFC intitulada *Internet Official Protocol Standards*. Essa RFC também inclui ponteiros para as RFCs mais atualizadas para vários padrões de protocolos. Já que as informações mudam freqüentemente, essa RFC é emitida novamente a cada 100 RFCs. O processo de padrões Internet em si é detalhado na RFC2026. Uma outra meta-RFC útil é a RFC2555, *30 Years of RFCs*, que descreve alguns dos contextos técnicos e culturais que existem por trás do sistema RFC.

Não se assuste pela opulência de detalhes técnicos encontrados nas RFCs. A maioria contém introduções, sumários e princípios úteis para os administradores de sistemas. Algumas são escritas especificamente como visões gerais ou introduções genéricas. As RFCs podem não ser a maneira mais suave de aprender um tópico, porém elas são autorizadas, concisas e gratuitas.

Nem todas as RFCs são austeras e cheias de detalhes técnicos maçantes. Algumas de nossas favoritas do tipo mais suave (normalmente escritas em 1.º de abril) são as RFCs 1118, 1149, 1925, 2324 e 2795:

- RFC1118 — *The Hitchhikers Guide to the Internet*
- RFC1149 — *A Standard for the Transmission of IP Datagrams on Avian Carriers*[2]
- RFC1925 — *The Twelve Networking Truths*
- RFC2324 — *Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0)*
- RFC2795 — *The Infinite Monkey Protocol Suite (IMPS)*

Elas se constituem numa excelente leitura e nos dão alguma idéia de quem são as pessoas que estão desenvolvendo e construindo nossa Internet.

Além de receber seu próprio número de série, uma RFC talvez também possa receber um número FYI (For Your Information, para sua informação), um número BCP (Best Current Practice, melhor prática atual) e um número STD (Standard, padrão). FYIs, STDs e BCPs são subséries das RFCs que incluem documentos de interesse ou importância especiais.

2. Um grupo de entusiastas Linux do BLUG (Bergen [Noruega] Linux Group), na realidade, implementou o CPIP (Carrier Pigeon Internet Protocol, protocolo Internet pombo-correio) conforme especificado na RFC1149. Para maiores detalhes, veja o site Web www.blug.linux.no/rfc1149!

www.snow666.blogspot.com
Snow666

As FYIs são documentos introdutórios ou informativos destinados ao público em geral. Elas normalmente são um excelente local para iniciar pesquisa sobre um tópico não familiar. As STDs documentam protocolos Internet que completaram os processos de revisão e de teste do IETF e foram adotados formalmente como padrões. As BCPs documentam procedimentos recomendados para sites Internet; elas são formadas por sugestões em termos de administração, e para os administradores de sistemas normalmente formam a série mais valiosa das subséries RFC.

RFCs, FYIs, STDs e BCPs são numeradas seqüencialmente dentro de suas próprias séries de modo que um documento pode originar vários números de identificação diferentes. Por exemplo, a RFC1635, *How to Use Anonymous FTP*, também é conhecida como FYI0024.

As RFCs são encontradas em diversas fontes. Há uma lista de espelhos mantidos ativamente em www.rfc-editor.org, que é a central de despacho para questões relacionadas com RFCs.

13.2 Mapa Rodoviário sobre Redes

Agora que já fornecemos a você um pouco de contexto, vamos dar uma olhada nos protocolos TCP/IP em si. TCP/IP é uma "família de protocolos", um grupo de protocolos de rede desenvolvido para funcionamento em conjunto de forma estável. Ele inclui vários componentes, cada um dos quais definido por uma RFC única ou por uma série delas:

- IP (Internet Protocol, protocolo Internet) que direciona pacotes de dados de uma máquina para uma outra (RFC791).
- ICMP (Internet Control Message Protocol, protocolo de mensagens de controle Internet) que fornece vários tipos de suporte de baixo nível para o IP, incluindo mensagens de erros, assistência de roteamento e ajuda para depuração (RFC792).
- ARP (Address Resolution Protocol, protocolo para determinação de endereços) que traduz endereços IP em endereços de hardware (RFC823)[3].
- UDP (User Datagram Protocol, protocolo de datagramas de usuário) e o TCP (Transmission Control Protocol, protocolo de controle de transmissão) que entregam dados para aplicativos específicos na máquina destinatária. O UDP dispõe de transporte não verificado e dentro das possibilidades para mensagens individuais enquanto o TCP garante uma conversação confiável, full-duplex, com controle de fluxo e com correção de erros entre os processos nos dois hosts (RFCs 768 e 793).

O TCP/IP é projetado em torno do esquema de camadas mostrado na Tabela 13.1.

Tabela 13.1 Modelo de rede TCP/IP.

Camada	Função
Aplicação	Programas aplicativos para o usuário final
Transporte	Entrega de dados aos aplicativos[a]
Link	Hardware de rede e drivers de dispositivos
Rede	Comunicação, endereçamento e roteamento básicos
Física	O cabo ou a mídia física em si

a. Opcionalmente questões relacionadas à confiabilidade de endereçamento e controle de fluxo.

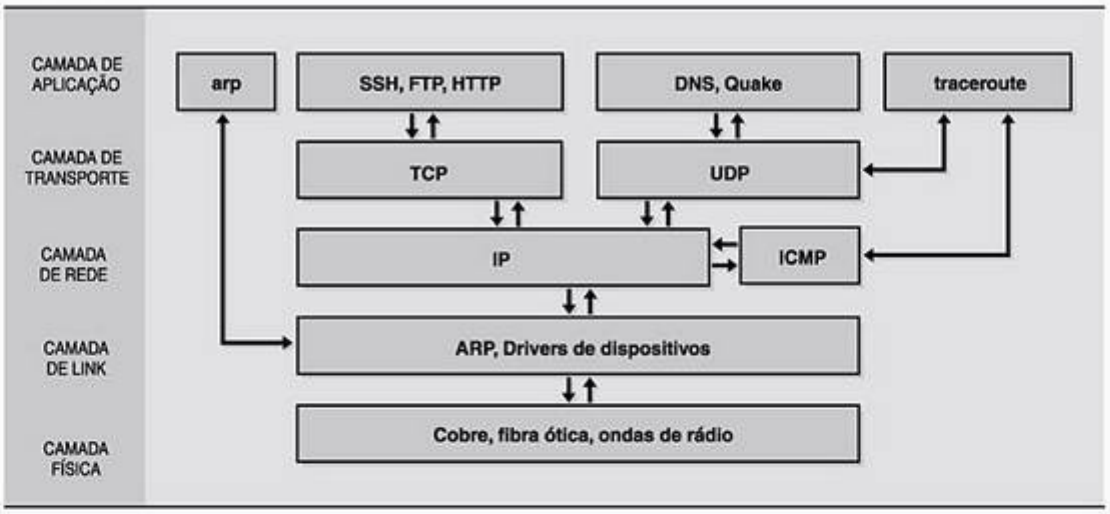
Depois de o TCP/IP ter sido implementado e empregado, a ISO (International Standards Organization, organização internacional de padrões) produziu sua própria família de protocolos de sete camadas chamada OSI. Ele era um elefante branco projetado por comitê consumado e jamais pegou realmente devido à sua complexidade e ineficiência. Alguns imaginam que uma camada financeira e uma camada política poderiam ser adicionadas ao modelo original OSI de sete camadas.[4]

O Diagrama A ilustra como os vários componentes e clientes do TCP/IP se encaixam nessa arquitetura geral e esquema de camadas.

3. Esta é na verdade, uma mentirinha. O ARP não faz parte realmente do TCP/IP e pode ser usado com outras famílias de protocolos. Entretanto, ele é parte integral da maneira pela qual o TCP/IP funciona na maioria das mídias para LANs.
4. Na realidade, uma camiseta exibindo esse modelo estendido de nove camadas se encontra disponível no Departamento de Ciências da Computação da Universidade de Colorado. Envie um e-mail para tshirt@cs.colorado.edu para obter maiores detalhes.

Snow666

Diagrama A Uma grande família TCP/IP feliz



13.3 Pacotes e Encapsulamento

O Linux pode suportar uma série de redes físicas, entre as quais Ethernet, FDDI, Token Ring, ATM (Asynchronous Transfer Mode, modo de transferência assíncrono), Ethernet sem fio e sistemas baseados em linha serial. O hardware é gerenciado dentro da camada de link da arquitetura TCP/IP, e protocolos de nível mais alto não sabem ou não dão importância hardware específico que está sendo usado.

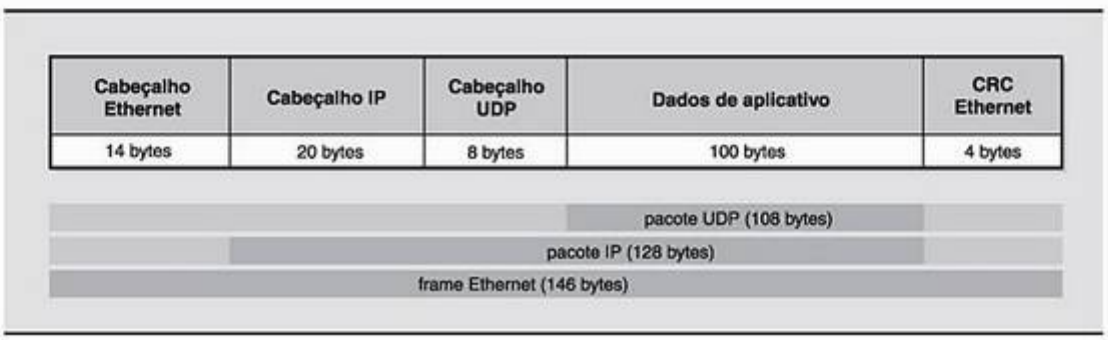
Os dados trafegam numa rede sob a forma de *pacotes*, seqüências de dados com um comprimento máximo imposto pela camada de link. Cada pacote é formado por um cabeçalho e uma carga útil. O cabeçalho informa de onde o pacote veio e para onde ele está indo. Ele também pode incluir checksums, informações específicas a um protocolo ou outras instruções de manipulação. A carga útil é formada pelos dados a serem transferidos.

O nome da unidade de dados primitiva depende da camada de protocolos. Na camada de link ela é chamada de *frame*, na camada IP de *pacote* e na camada TCP de *segmento*. Aqui, usamos pacote como um termo genérico que engloba todos esses casos.

À medida que o pacote trafega para baixo dentro da pilha de protocolos (do transporte TCP ou UDP para IP para Ethernet para o cabo físico) em preparação para ser enviado, cada protocolo adiciona sua própria informação de cabeçalho. Cada pacote finalizado do protocolo se torna a parte carga útil do pacote gerado pelo próximo protocolo. Esse aninhamento é conhecido como encapsulamento. Na máquina receptora, o encapsulamento é invertido à medida que o pacote trafega de volta para cima pela pilha de protocolos.

Por exemplo, um pacote UDP sendo transmitido por rede Ethernet contém três envoltórios ou envelopes diferentes. Na rede Ethernet, lhe é aplicado um "frame" com um cabeçalho simples que lista a fonte e os endereços de hardware de destino do próximo salto, o comprimento do frame e o checksum do frame (CRC). A carga útil do frame Ethernet é um pacote IP, a carga útil do pacote IP é um pacote UDP e a carga útil do pacote UDP é formada pelos dados em si que estão sendo transmitidos. O Diagrama B ilustra os componentes de tal frame.

Diagrama B Um pacote de rede típico



Snow666

Usamos o termo "byte" para nos referirmos a uma unidade de dados de 8 bits. No passado, "byte" era um termo mais genérico, de modo que freqüentemente você verá o termo "octeto" usado em seu lugar nas RFCs.

A camada de link

Nesta seção, cobrimos vários tópicos que ligam o intervalo entre as camadas mais baixas do software de rede e o hardware de rede em si.

Padrões de framing Ethernet

Uma das principais tarefas da camada de link é adicionar cabeçalhos a pacotes e de colocar separadores entre eles. Os cabeçalhos contêm os checksums e as informações de endereçamento da camada de link dos pacotes, e os separadores garantem que os receptores consigam informar onde um pacote termina e onde o próximo se inicia. O processo de adicionar esses trechos de informação é conhecido genericamente como framing.

Dois padrões distintos para framing Ethernet de 10 Mb/s são usados comumente: DIX Ethernet II e o IEE 802.3 LLC SNAP.[5] Os hosts Linux usam Ethernet II, como faz a maioria dos hosts UNIX e roteadores Cisco. As redes Novell ou IPX e os hosts Windows normalmente usam 802.2. Ethernet II e 802.2 diferem em alguns campos do cabeçalho do frame, porém não conflitam, de modo que os receptores podem determinar de forma não ambígua qual formato está sendo usado por cada pacote individual e decodificar o cabeçalho de maneira apropriada.

O framing que uma máquina usa é determinado tanto pela sua placa de rede quanto pelo driver da placa de rede. Em PCs rodando Windows podemos escolher que estilo de framing desejamos, porém no Linux normalmente isto não é possível. Ambos os tipos de framing operam perfeitamente entre si sob a perspectiva do Linux. Por outro lado, as máquinas Windows que usam framing distinto na mesma rede não conseguem se comunicar entre si. Na qualidade de administrador de sistemas, você não precisa se preocupar com discordâncias de framing a menos que esteja efetuando uma depuração de baixo nível numa rede mista.

Cabeamento Ethernet e padrões de sinalização

As opções de cabeamento para as diversas velocidades Ethernet (10 Mb/s, 100 Mb/s, 1 Gb/s e agora 10 Gb/s) normalmente são especificadas como parte da iniciativa de padronização do IEEE. Normalmente, um único tipo de cabo com limites de distância curta será aprovado à medida que uma nova tecnologia emergir. Tipos de mídia mais baratos e limites mais generosos serão acrescentados posteriormente.

Veja o o Capítulo 15, *Hardware de Rede*, para obter maiores informações sobre os diversos tipos de padrão. Uma outra referência útil para os prós e contras da Ethernet é o site Web wwwhost.ots.utexas.edu/ethernet que é mantido por Charles Spurgeon.

Redes sem fio

O padrão IEEE 802.11 tenta definir os padrões de framing e de sinalização para redes sem fio. Infelizmente, ele era originalmente um tanto vago e incluía vários parâmetros e opções que não foram especificados completamente. Um problema de interoperabilidade que talvez você precise prestar atenção é aquele da "tradução" *versus* "encapsulamento".

A tradução converte um pacote de um formato em outro; o encapsulamento envolve o pacote com o formato desejado. Os sistemas Windows tendem a usar como padrão o encapsulamento e os sistemas Linux, a tradução; as estações-base sem fio têm de estar configuradas explicitamente. Se você estiver empregando uma rede sem fio, terá de ter certeza que suas estações-base e as estações de trabalho com as quais elas conversam estejam todas operando no mesmo modo.

MTU

O tamanho dos pacotes em uma rede pode ser limitado tanto pelas especificações de hardware quanto pelas convenções dos protocolos. Por exemplo, a carga útil de um frame Ethernet padrão não pode ultrapassar os 1.500 bytes. O limite de tamanho está associado ao protocolo da camada de link e é chamado de MTU (Maximum Transfer Unit, unidade de transferência máxima). A Tabela 13.2 ilustra valores típicos para o MTU.

---

5. A camada de link é, na verdade, dividida em duas partess: MAC, a subcamada Media Access Control, ou seja, controle de acesso à mídia, e a LLC, a subcamada Link Layer Control, ou seja, controle de camada de link. A camada MAC lida com a mídia e coloca os pacotes nos cabos. A camada LLC trata do framing.

Snow666

Tabela 13.2 MTUs para diversos tipos de camada de link de rede.

Tipo de rede	Unidade de transferência máxima
Ethernet	1.500 bytes (1.492 com framing 802.2)
FDDI	4.470 bytes (4.352 para IP/FDDI)
Token ring	Configurável[a]
Link de modem PPP	Configurável, normalmente 512 ou 576 bytes
Pilhas de Pc	Configurável, normalmente o padrão é de 512
Links WAN ponto-a-ponto (T1, T3)	Configurável, normalmente 1.500 ou 4.500 bytes

a. Valores comuns são 552; 1.064; 2.088; 4.508 e 8.232. Algumas vezes 1.500 para atender ao padrão Ethernet.

Na família TCP/IP, a camada IP divide pacotes para atender ao MTU de um link de rede em particular. Se um pacote é roteado através de diversas redes, uma das redes intermediárias pode ter um MTU menor do que a rede de origem. Nesse caso, o roteador que encaminha o pacote para a rede de menor MTU subdividirá ainda mais o pacote num processo denominado fragmentação. A fragmentação é uma tarefa malvista por um roteador atarefado. O protocolo TCP é capaz de determinar o menor MTU ao longo do caminho para o destino e usar este tamanho desde o princípio. O UDP não é tão bom e fica contente por passar esse trabalho extra para a camada IP. No IPv6, um novo protocolo trilha seu caminho através do processo de padronização, os roteadores intermediários não podem mais executar a fragmentação; é necessário descobrir o MTU.

Problemas de fragmentação podem ser insidiosos. Se estiver, por exemplo, usando uma arquitetura com túneis para uma rede privada virtual, você terá de analisar o tamanho dos pacotes que estão atravessando o túnel. Normalmente eles têm 1.500 bytes só para começar, mas assim que o cabeçalho de tunelamento for adicionado, eles passam a ter por volta de 1.540 bytes e têm de ser fragmentados. Configurar o MTU do link para um valor menor evitará a fragmentação e aumentará o desempenho global da rede com tunelamento. Consulte a página de manual **ifconfig** para ver como configurar o MTU de uma interface.

Endereçamento de pacotes

Assim como as cartas ou mensagens de e-mail, os pacotes de rede têm de ser endereçados apropriadamente para chegarem a seus destinos. São usados vários esquemas de endereçamento combinados:

- Endereços MAC (Medium Access Control, controle de acesso à mídia) para hardware
- Endereços IP para software
- Nomes de host para as pessoas

A interface de rede de um host normalmente possui um endereço MAC de camada de link que a distingue de outras máquinas na mesma rede física, um endereço IP que a identifica na Internet global e um nome de host que é usado pelos seres humanos.

O nível mais baixo de endereçamento é ditado pelo hardware da rede. Por exemplo, dispositivos Ethernet recebem um endereço de hardware exclusivo de 6 bytes no momento da fabricação.[6] As interfaces Token Ring possuem um endereço similar que também tem 6 bytes de comprimento. Algumas redes ponto-a-ponto (como PPP, descritas na seção 13.13) não precisam de nenhum endereço de hardware; a identidade do destino é especificada quando o link é estabelecido.

Um endereço Ethernet de 6 bytes é dividido em duas partes: os primeiros três bytes identificam o fabricante do hardware e os três últimos formam um número de série exclusivo que o fabricante atribui. Os administradores de sistemas normalmente conseguem identificar pelo menos a marca do dispositivo que está abarrotando a rede por meio de observação do identificador de 3 bytes numa tabela de IDs de fornecedores. Há uma tabela dessas atualizada no endereço a seguir:

http://www.iana.org/assignments/ethernet-numbers

6. Exclusivos pelo menos na teoria. Certa vez a 3Com duplicou números Ethernet entre as suas placas com diferentes tipos de conectores de rede; eles supuseram que os clientes fariam pedidos somente de um único tipo. Esse atalho gerou grande tumulto e problemas em instalações que estavam numa fase de transição entre mídias e chegou até mesmo a provocar problemas na própria rede interna da 3Com. Conflitos de endereços MAC são fatais na mesma rede, porém são aceitáveis em redes separadas por um roteador.

www.snow666.blogspot.com
Snow666

Essas informações costumavam ser publicadas regularmente nas séries RFC, mas não são mais distribuídas desta maneira. A RFC1700 (1994) foi a última RFC *Assigned Numbers*, isto é, com números atribuídos. O repositório oficial de todos os números mágicos da Internet agora é www.iana.org/numbers.htm.

Endereços de hardware de interfaces Ethernet devem ser atribuídos permanentemente e de maneira imutável; infelizmente, algumas placas de rede permitem que especifiquemos o endereço de hardware. Placas sem fio são especialmente problemáticas nesse aspecto. Não atribua valores no intervalo de endereços multidifusão (segundo dígito ímpar) nem use outros valores especiais.

O Linux permite que alteremos o endereço de hardware da interface Ethernet, porém, por favor, não faça isto; você poderá desmontar firewalls e algumas implementações DHCP. Entretanto, esse recurso pode ser útil caso você tenha de substituir uma máquina ou placa de rede danificada e, por alguma razão, tenha que usar o endereço MAC antigo (por exemplo, todos os seus comutadores o filtram ou seu servidor DHCP distribui endereços baseados em endereços MAC).

No próximo nível acima do hardware, é usado o endereçamento Internet (mais comumente conhecido como endereçamento IP). Tipicamente, é atribuído um[7] endereço IP de 4 bytes a cada interface de rede. Os endereços IP são globalmente exclusivos[8] e independentes de hardware. Continuamos falando sobre endereços IP na próxima seção.

Veja a seção 13.6 para obter maiores informações sobre ARP.

A associação (mapeamento) entre endereços IP e endereço de hardware é implementada na camada de link do modelo TCP/IP. Em redes que suportam difusão ampla (por exemplo, redes que permitem que pacotes sejam endereçados a "todos os hosts nesta rede física"), o protocolo ARP permite que essas associações sejam descobertas automaticamente, sem a ajuda de um administrador de sistemas.

Já que os endereços IP são longos, aparentemente números aleatórios, eles são difíceis para as pessoas se lembrarem. Os sistemas Linux permitem que um ou mais nomes de hosts sejam associados a um endereço IP, de modo que os usuários podem digitar yahoo.com em vez de 216.115.108.245. Essa associação pode ser configurada de várias maneiras, indo desde um arquivo estático (**/etc/hosts**) até o sistema de banco de dados NIS ou DNS (Domain Name System), o sistema mundial para nomes de domínios. Tenha em mente que os nomes de hosts são, na realidade, apenas uma abreviatura conveniente dos endereços IP.

Portas

Os endereços IP identificam máquinas, ou mais precisamente, interfaces de rede em uma máquina. Eles não são específicos o bastante para endereçar processos ou serviços particulares. O TCP e o UDP estendem os endereços com um conceito conhecido como "porta". Uma porta é um número de 16 bits que suplementa um endereço IP para especificar um determinado canal de comunicação. Serviços-padrão como e-mail, FTP e a Web se associam a portas "bem conhecidas" definidas em **/etc/services**. Para ajudar a evitar a apropriação indevida desses serviços, os sistemas Linux restringem os programas de servidor de se vincularem a números de porta abaixo de 1.024, a menos que eles estejam rodando como root. (Mas qualquer um pode se comunicar com um servidor que está rodando num número de porta baixo; a restrição se aplica somente a assumir o controle da porta.)

Tipos de endereço

Tanto a camada IP quanto a camada de link definem vários tipos diferentes de endereços:

- Unidifusão — endereços que se referem a um único host (interface de rede, na verdade)
- Multidifusão — endereços que identificam um grupo de hosts
- Difusão ampla — endereços que incluem todos os hosts da rede local

O endereçamento multidifusão facilita aplicativos como videoconferência no qual o mesmo conjunto de pacotes tem de ser enviado a todos os participantes. O IGMP (Internet Group Management Protocol, protocolo de gerenciamento de grupos Internet) constrói e gerencia conjuntos de hosts que são tratados como um único destino multidifusão. A multidifusão ainda é algo experimental. Entretanto, os mercados de voz por meio de IP e de vídeo sob demanda podem acelerar um pouco o seu emprego.

Os endereços IP multidifusão começam com um byte dentro do intervalo de 224 a 239. Os endereços atingem todos os hosts da rede local usando um tipo de endereço-curinga especial na qual a representação binária da parte host (definida a seguir) é formada completamente por 1s.

---

7. As interfaces de rede podem, na verdade, ter mais do que um endereço IP associado a elas, porém, se trata de uma configuração especial que é usada somente em algumas circunstâncias específicas. Veja *Configurando Interfaces Virtuais* na página 522 para obter maioers informações.
8. Essa é uma pequena mentira que é verdade na maioria dos casos. Leia a discussão sobre o NAT que se inicia um pouco mais à frente, neste capítulo, para obter as informações exatas sobre endereços IP não exclusivos.

Snow666

13.4 Endereços IP: Os Detalhes Viscerais

Um endereço IP ou Internet tem quatro bytes de comprimento e é dividido numa parte rede e uma parte host. A parte rede identifica uma rede lógica à qual o endereço se refere e a parte host identifica uma máquina nesta rede.

Por convenção, os endereços IP são escritos como números decimais, um para cada byte, separados por pontos. Por exemplo, o endereço IP para nossa máquina "boulder" é escrito como 128.138.240.1. O byte mais à esquerda é o mais significativo e sempre faz parte da porção rede.

Quando 127 é o primeiro byte de um endereço, ele representa a "rede loopback", uma rede fictícia que não possui nenhuma interface de hardware real e somente um host. O endereço loopback 127.0.0.1 sempre se refere ao host atual. Seu nome simbólico é "localhost".

O endereço IP de uma interface e outros parâmetros são configurados pelo comando **ifconfig**.

Pule para a seção 13.7 para obter uma descrição detalhada de **ifconfig**.

Classes de endereços IP

Historicamente, os endereços IP eram agrupados em classes, dependendo dos primeiros bits do byte mais à esquerda. A classe determinava quais bytes do endereço se encontravam na porção rede e quais na porção host. Hoje, os sistemas de roteamento usam uma máscara explícita para especificar a porção rede e são capazes de definir limites entre quaisquer dois bits, não somente nos limites dos bytes. Entretanto, as classes tradicionais ainda são usadas como padrão quando nenhuma divisão explícita for fornecida.

As classes A, B e C representam endereços IP regulares. As classes D e E são usadas para multidifusão e fins de pesquisa. A Tabela 13.3 descreve as características de cada uma das classes.

A porção rede de um endereço é representada por N e a porção host, por H.

Tabela 13.3 Classes de endereços Internet históricas.

Classe	1º byte[a]	Formato	Comentários
A	1-126	N.H.H.H	As primeiras redes ou reservadas para DOD.
B	128-191	N.N.H.H	Grandes instalações, normalmente com sub-redes, eram difíceis de ser conseguidas.
C	192-223	N.N.N.H	Fáceis de ser conseguidas, normalmente obtidas em conjuntos.
D	224-239	–	Endereços multidifusão, atribuídos de uma forma não permanente.
E	240-255	–	Endereços experimentais.

a. O valor 0 é especial e não é usado como primeiro byte de endereços IP comuns. 127 é reservado para o endereço de loopback.

Sub-redes e máscaras de rede

É raro para uma única rede física ter mais do que uma centena de computadores conectados a ela. Portanto, os endereços classe A e classe B (que permitem, respectivamente, até 16.777.214 e 65.534 hosts por rede) são, na realidade, estúpidos e um desperdício. Por exemplo, as 126 redes classe A usam até metade do espaço de endereços disponível.

Instalações com esses endereços usam um refinamento do esquema de endereçamento, chamado sub-redes, no qual parte da porção host de um endereço é "emprestada" para estender a porção de rede. Por exemplo, os quatro bytes de um endereço classe B normalmente seriam interpretados como N.N.H.H. Se for empregado o recurso de sub-rede para atribuir o terceiro byte para o número de rede em vez do número de host, o endereço seria interpretado como N.N.N.H.

Esse emprego de sub-redes transforma um único endereço de rede classe B em 256 redes distintas tipo classe C, cada uma das quais capaz de suportar até 254 hosts.

*Consulte a seção 13.7 para obter maiores informações sobre **ifconfig**.*

Essa nova atribuição é efetuada pelo emprego do comando **ifconfig** para associar uma "máscara de sub-rede" explícita a uma interface de rede. Cada bit da máscara de rede que corresponde à porção rede de um endereço IP é configurado em 1 e os bits host são configurados em 0. Por exemplo, a máscara de rede para a configuração N.N.N.H seria 255.255.255.0 em decimal ou 0xFFFFFF00 em hexadecimal. **ifconfig** normalmente usa a classe inerente de um endereço para descobrir que bits fazem parte da rede. Ao configurarmos uma máscara explícita, simplesmente anulamos esse comportamento.

A divisão entre a parte rede e a parte host não precisa cair num limite de byte. Entretanto, os bits de rede têm de ser contíguos e aparecer no lado da ordem mais alta do endereço.[9]

9. Configurações como N.N.H.N já foram permitidas no passado, mas eram incomuns; hoje elas não são mais permitidas.

Snow666

As máscaras de sub-rede que não terminam num limite de byte podem ser irritantes para serem decodificadas e normalmente são escritas como /XX, onde XX é o número de bits na porção de rede do endereço. Isso algumas vezes é chamado de notação CIDR (Classless Inter-Domain Routing, roteamento entre domínios sem classes). Por exemplo, o endereço de rede 128.138.243.0/26 refere-se à primeira de quatro redes cujos primeiros bytes são 128.138.243. As outras três redes têm 64, 128 e 192 como seus primeiros quatro bytes. A máscara de sub-rede associada a essas redes é 255.255.255.192 ou 0xFFFFFFC0; em binário, são 26 números 1 seguidos por 6 zeros. O Diagrama C ilustra os relacionamentos entre esses números com um pouco mais de detalhe.

Diagrama C Conversão de base de máscara de sub-rede.

Endereço IP	128 .	138 .	243 .	0
Máscara de rede decimal	255 .	255 .	255 .	192
Máscara de rede hexadecimal	f f .	f f .	f f .	c 0
Máscara de rede binária	1111 1111 .	1111 1111 .	1111 1111 .	1100 0000

Uma rede /26 possui 6 bits deixados (32 – 26 = 6) para numerar hosts. 2^6 é 64, de modo que a rede possui 64 endereços de host possíveis. Porém, ela é capaz, efetivamente, de acomodar apenas 62 hosts, pois os endereços de host formados somente com 0s ou somente com 1s são reservados (eles são, respectivamente, os endereços de rede e de difusão ampla).

É confuso fazer toda essa conversão de bits de cabeça, porém alguns truques poderão torná-la mais simples. O número de hosts por rede e o valor do último byte na máscara de rede sempre dão no total 256:

último byte da máscara de sub-rede = 256 – tamanho da rede

Por exemplo, 256 – 64 = 192, que é o byte final da máscara de sub-rede do exemplo anterior. Um outro fato aritmético é que o último byte de um verdadeiro endereço de rede (ao contrário de uma máscara de sub-rede) tem de ser divisível exatamente pelo número de hosts por rede. Vemos isso em ação no exemplo presente, onde os últimos bytes são 0, 64, 128 e 192 — todos divisíveis exatamente por 64.

Em nosso exemplo, os dois bits extras do endereço de rede obtidos por aplicação de sub-redes podem assumir os valores 00, 01, 10 e 11. A rede 128.138.243.0/24 foi dividida então em quatro redes /26.

- 128.138.243.0/26 (0 em decimal é **00**000000 em binário)
- 128.138.243.64/26 (64 em decimal é **01**000000 em binário)
- 128.138.243.128/26 (128 em decimal é **10**000000 em binário)
- 128.138.243.192/26 (192 em decimal é **11**000000 em binário)

Os bits em negrito do último byte de cada endereço são os bits pertencentes à porção rede deste byte.

Dado um endereço IP (digamos, 128.138.243.100), não podemos dizer sem o emprego da máscara de rede associada quais serão os endereços de rede e de difusão ampla. A Tabela 13.4 ilustra as possibilidades para /16 (o padrão para um endereço classe B), /24 (um valor sensível) e /26 (um valor realista caso o espaço de endereços seja apertado).

Tabela 13.4 Exemplo de decodificação de endereços IP.

Endereço IP	Máscara de rede	Rede	Difusão ampla
128.138.243.100/16	255.255.0.0	128.138.0.0	128.138.255.255
128.138.243.100/24	255.255.255.0	128.138.243.0	128.138.243.255
128.138.243.100/26	255.255.255.192	128.138.243.64	128.138.243.127

Keith Owens criou um pequeno e maravilhoso script em Perl chamado **ipcalc.pl** que ajuda nesta aritmética binária/hexadecimal/máscara. Ele se encontra disponível em ftp.ocs.com.au e requer Perl 5. **ipcalc** exibe tudo que você precisaria saber sobre um endereço de rede e respectivos endereços de difusão ampla, máscara de rede, hosts, etc. Até encontramos uma versão de **ipcalc** que foi portada para Palm Pilot; vá ao endereço www.ajw.com/ipcalc.htm.

Snow666

Eis uma amostra da saída gerada por ipcalc, modificada um pouco para ajudar na formatação:

```
% ipcalc.pl 128.138.243.100/26
IP address    128 . 138 . 243 . 100 / 26 128.138.243.100/26
Mask bits   11111111 11111111 11111111 11000000
Mask bytes    255 . 255 . 255 . 192         255.255.255.192
Address     10000000 10001010 11110011 01100100
Network       128 . 138 . 243 . 64          128.138.243.64
Broadcast     128 . 138 . 243 . 127         128.138.243.127
First Host    128 . 138 . 243 . 65          128.138.243.65
Last Host     128 . 138 . 243 . 126         128.138.243.126
Total Hosts         62
PTR                 100.243.138.128.in-addr.arpa
IP Address (hex)    808AF364
```

A saída fornece tanto versões fáceis de serem compreendidas dos endereços quanto versões "cortar e colar". Legal.

O Red Hat inclui um programa, também chamado **ipcalc**, que é muito incompleto e para a maioria dos cálculos supõe que os endereços IP estejam nas classes A, B ou C.

A RFC original sobre aplicação de sub-redes em endereços IP (RFC950) não permite o emprego da primeira ou da última sub-redes (tudo 0 e tudo 1). Em nosso exemplo com as redes /26, essa regra eliminaria metade das sub-redes: a sub-rede 0 e a sub-rede 192. Todo o mundo ignorou a RFC, exceto a Novell e a Cisco. (Em versões antigas do sistema operacional IOS da Cisco, tínhamos de habilitar explicitamente a sub-rede 0 por meio do comando **ip subnet zero**. Nas versões 12.0 e posteriores, a sub-rede 0 é disponível como padrão.)

A RFC está enganada, embora tenha boas intenções. A sub-rede 0 não foi permitida, pois se pensava que poderia surgir confusão caso o endereço de sub-rede fosse indistinguível de um endereço de rede em que não tivesse sido aplicado a sub-rede. Porém, esse temor provou-se sem sentido e as sub-redes tudo 0/tudo 1 são usadas comumente hoje em dia. É a porção host que não deveria ser tudo 0 ou tudo 1.

O endereço de rede e o endereço de difusão ampla roubam dois hosts de cada rede, de modo que pareceria que a menor rede significativa teria quatro hosts possíveis: dois hosts reais — usualmente em uma das extremidades de um **link** ponto-a-ponto — e os endereços de rede e de difusão ampla. Ter quatro valores para hosts requer dois bits na porção host, de modo que uma rede destas seria uma rede /30 com máscara de rede 255.255.255.252 ou 0xFFFFFFFC. Entretanto, uma rede /31 é tratada como um caso especial (veja a RFC3021) e não possui nem endereço de rede nem endereço de difusão ampla; seus dois endereços são usados para hosts e sua máscara de rede é 255.255.255.254.

Embora os hosts numa rede possam concordar que estão usando endereços nos quais se aplicou sub-rede, o resto do mundo não sabe a este respeito e continua a tratar endereços de acordo com suas classes implícitas.[10] Em vez de anunciar cada sub-rede para o mundo externo, em nosso exemplo 128.138.243.100 você precisaria anunciar somente uma única rede classe B. Assim que um pacote chegasse dentro da área onde foi aplicada a sub-rede, seu endereço de destino seria interpretado novamente com máscaras de rede locais, a verdadeira rede de destino seria "descoberta", e o pacote encaminhado para o seu destino exato.

A crise de endereços IP

A comunidade da Internet se deu conta por volta de 1992 que havia três problemas fundamentais com o esquema de alocação de endereços original:

- Primeiramente, iríamos ficar sem endereço classe B — os mais desejáveis para organizações moderadamente grandes — na metade do ano de 1995.
- Em segundo lugar, as tabelas de roteamento de sites backbone Internet estavam crescendo a tal ponto de não caberem mais na memória dos roteadores disponíveis na época.
- Por fim, os endereços IP eram alocados de uma maneira em que o primeiro que chegasse era o primeiro a ser atendido sem nenhuma localidade de referência; isto é, endereços numericamente adjacentes poderiam estar na mesma organização ou em continentes diferentes. Imagine a confusão que resultaria caso os números de telefone ou códigos de endereçamento postal fossem atribuídos deste modo indiscriminado.

10. Uma outra mentira no nome de uma descrição simples, bem como ainda incompleta; veja a discussão sobre CIDR (Classless Inter-Domain Routing, roteamento entre domínios sem classes) na seção seguinte para se inteirar sobre o verdadeiro escopo.

www.snow666.blogspot.com
Snow666

Para resolver o problema, foram apresentadas duas soluções associadas: uma para o futuro próximo e outra a longo prazo. O CIDR (Classless Inter-Domain Routing, roteamento entre domínios sem classes), a solução de curto prazo, é uma maneira diferente de gerenciar o espaço de endereços de quatro bytes existente que utiliza os endereços disponíveis de maneira mais eficiente e permite que tabelas de roteamento sejam simplificadas levando em conta adjacências numéricas. Discutimos o CIDR de forma mais detalhada na próxima seção.

A solução a longo prazo, IPv6, é uma revisão do protocolo IP que expande o espaço de endereços para 16 bytes e incorpora várias outras lições aprendidas através do uso do IP ao longo dos últimos 25 anos. Ele elimina vários recursos do IP que a experiência demonstrou serem de pouca valia, tornando o protocolo potencialmente mais rápido e mais fácil de ser implementado. Ele também integra recursos de segurança e de autenticação no protocolo básico e elimina a fragmentação em roteadores intermediários.

O IPv6 ainda se encontra nos estágios iniciais de desenvolvimento, porém o CIDR está totalmente operacional há anos. O CIDR é suportado e usado pelo backbone Internet e pela maioria dos fabricantes de equipamentos para roteamento. O NAT (Network Address Translation, tradução de endereços de rede), um esquema para reutilizar endereços IP que é visto um pouco mais à frente, neste capítulo, também desempenhou um grande papel na redução da demanda por endereços IP.

A complexidade do IPv6, a eficiência do CIDR e do NAT e a inércia de uma Internet que já funciona relativamente bem são todos fatores que se combinam para sugerir que pode levar um bom tempo antes de passarmos para o IPv6, se de fato algum dia o fizermos. Tal mudança provavelmente será dirigida por países como Japão ou China que não conseguem obter o espaço de endereços IPv4 de que eles precisam ou por uma nova aplicação avassaladora que requeira o IPv6. Um bom candidato a tal aplicativo seria uma nova geração de telefones celulares e outros dispositivos sem fio que possam ter cada um deles um endereço IPv6 que faria a associação a um número de telefone. Sistemas de voz sobre IP também se beneficiariam de uma correspondência mais próxima entre números de telefone e endereços IPv6.

Alguns detalhes adicionais sobre endereçamento IPv6 são dados ainda nesta seção.

CIDR

O CIDR (Classless Inter-Domain Routing, roteamento entre domínios sem classes), definido na RFC1519 (setembro de 1993), elimina o sistema de classes que outrora determinava a porção rede de um endereço IP. Da mesma forma que sub-redes, das quais ele é uma extensão direta, ele depende de uma máscara de rede explícita para definir o limite entre as partes rede e host de um endereço. Mas diferentemente da sub-rede, ele permite, para fins de roteamento, que a porção rede seja menor do que seria implícito pela classe implícita de um endereço. Usar uma máscara de rede mais curta possui o efeito de agregar diversas redes. Portanto, o CIDR é algumas vezes chamado de super-rede.

Com o CIDR, várias redes classe C podem ser alocadas a uma instalação sem exigir que a Internet tenha entradas de tabelas de roteamento distintas para cada uma delas. A instalação também poderia ter alocado um subespaço de um endereço classe A ou B. Suponha, por exemplo, que tenha sido concedido a uma instalação um bloco de 8 endereços classe C numerados de 192.144.0.0 a 192.144.7.0 (em notação CIDR, 192.144.0.0/21). Internamente, a instalação poderia usá-los como

- 1 rede de comprimento /21, 2.046 hosts[11], máscara de rede 255.255.248.0
- 8 redes de comprimento /24, 254 hosts cada, máscara de rede 255.255.255.0
- 16 redes de comprimento /25, 126 hosts cada, máscara de rede 255.255.255.128
- 32 redes de comprimento /26, 62 hosts cada, máscara de rede 255.255.255.192 e assim por diante. Também é possível misturarmos e combinarmos regiões de comprimentos de sub-rede diferentes, desde que todos os trechos se juntem sem sobreposição. Isso é chamando de sub-rede de comprimento variável. Por exemplo, um ISP com a alocação 192.144.0.0/21 poderia definir cerca de /30 redes para clientes PPP discados, cerca de /24 redes para clientes grandes e cerca de /27 redes para clientes menores.

Ao misturarmos e combinarmos dessa forma, todos os hosts de uma determinada rede terão de ser configurados com a mesma máscara de rede. Não podemos informar um host da rede que ele é um /24 e um outro host nessa mesma rede que ele é um /25.

A beleza e o valor do CIDR está no fato de que, sob a perspectiva da Internet, não é necessário ter 256, 128 ou até mesmo 32 entradas de tabelas de roteamento para estes endereços. Todos eles se referem à mesma organização e todos os pacotes precisam ir para o mesmo lugar. Uma única entrada de roteamento para o endereço 192.144.0.0/21

11. O padrão Ethernet original em cabos coaxiais RG-11 permitia no máximo 1.024 hosts numa única rede; com os comutadores atuais é possível (porém não muito sensato) realmente construirmos redes imensas.

Snow666

já basta. Além disso, o CIDR facilita a alocação de porções de endereço classe A e B e, conseqüentemente, aumenta em muitas vezes o número de endereços disponíveis.

Com o advento do CIDR, os administradores de sistemas se tornaram bons em aritmética binária e hexadecimal ou então descobriram que o utilitário **bc** do Linux é capaz de realizar cálculos matemáticos em qualquer base, usando as diretivas **ibase** e **obase**.[12] Você pode utilizar a Tabela 13.5 como uma referência.

Tabela 13.5 Configurações para vários comprimentos de máscara de rede.

Comprimento[a]	Bits de host	Hosts/rede[b]	Máscara de sub-rede decimal	Máscara de sub-rede hexadecimal
/20	12	4094	255.255.240.0	0xFFFFF000
/21	11	2046	255.255.248.0	0xFFFFF800
/22	10	1022	255.255.252.0	0xFFFFFC00
/23	9	510	255.255.254.0	0xFFFFFE00
/24	8	254	255.255.255.0	0xFFFFFF00
/25	7	126	255.255.255.128	0xFFFFFF80
/26	6	62	255.255.255.192	0xFFFFFFC0
/27	5	30	255.255.255.224	0xFFFFFFE0
/28	4	14	255.255.255.240	0xFFFFFFF0
/29	3	6	255.255.255.248	0xFFFFFFF8
/30	2	2	255.255.255.252	0xFFFFFFFC

a. O comprimento de rede + o número de bits de host é sempre 32, já que estamos dividindo a "torta" de tamanho fixo num endereço IP de 32 bits.

b. O pessoal fanático por matemática perceberá que o numero de hosts por rede é igual a (2^#bitshost – 2); o –2 reflete o fato de os endereços de host tudo 0 e tudo 1 serem especiais. A notação (2^#bitshost –2) significa 2 elevado ao número de bits, no campo hosts menos dois.

Quando o CIDR foi introduzido em 1993, as tabelas backbone continham aproximadamente 20.000 rotas. Apesar do crescimento exponencial da Internet desde então, o tamanho das tabelas de roteamento cresceu para somente 100.000 rotas no terceiro trimestre de 2001. Esse crescimento modesto nas entradas de roteamento se deve à extensiva agregação tanto de alocações de endereços antigas quanto novas.[13]

Uma região desagregada do espaço de endereços, chamada de pântano 192 (e pântanos menores nos intervalos 199 e 205), é formada por endereços classe C antigos cujos proprietários não os podem agregar e não querem entregá-los e depois ter que renumerá-los. Os Estados Unidos são particularmente ruins nesse aspecto. A Europa e a Ásia, que começaram um pouco mais tarde, aprenderam com os erros dos norte-americanos e fizeram um trabalho de alocação de endereços muito melhor.

Instalações com uma rede 192 desagregada deveriam devolvê-la para o ARIN (American Registry for Internet Numbers, registro americano para números Internet) e receber um novo bloco de seus ISPs. Infelizmente, o custo de renumeração (no espaço IPv4, pelo menos) impede que a maioria das instalações faça isso.

Embora o CIDR tenha sido imaginado apenas como uma solução interina, ele demonstrou ser sólido o bastante para lidar com os problemas de crescimento da Internet num futuro visível. De fato, o CIDR se deu tão bem que se tornou pouco claro se realmente precisamos de um novo protocolo IP. Uma enorme quantidade de trabalho de engenharia foi consumida pela especificação IPv6 e em implementações e ambiente de produção do IPv6. Seria uma pena perder todo esse trabalho, mas o emprego no atacado do IPv6 provavelmente irá exigir uma nova aplicação "matadora" escrita somente para o IPv6, um brinquedo feito para "executivos" que use o endereçamento IPv6 ou então uma decisão por parte da Microsoft em tornar obsoleto o IPv4.

12. Porém, tome cuidado para não se encurralar ... Este quebra-cabeça é deixado como exercício para o leitor.
13. As tabelas de roteamento estavam crescendo num ritmo horripilante ao longo de 2000, porém o crescimento desacelerou significativamente desde então, provavelmente devido à situação econômica. O hardware e algoritmos de roteamento atuais são capazes de sustentar um crescimento de alguns pontos percentuais ao ano, mas não os 25% que estávamos presenciando na época de euforia da indústria .com. Veja novo endereço bgp.potaroo.net para obter maiores informações.

Snow666

Alocação de Endereços

Nos primórdios da Internet, sites individuais requisitavam espaço de endereço à InterNIC (Network Information Center, centro de informações de rede). O ARIN agora substituiu a InterNIC nas Américas.[14] Somente ISPs que alocam quantidades significativas de espaço de endereço por ano estão eleitos a requisitar espaço de endereços IP ao ARIN. Todos os demais sites têm de fazer solicitação a seus ISPs.

Somente números de rede são atribuídos formalmente; os sites têm que definir seus próprios números de host para formarem endereços IP completos. Você pode subdividir o espaço de endereços dado a você em sub-redes caso prefira.

Administrativamente, o ICANN (Internet Corporation for Assigned Names and Numbers) delegou os blocos de endereços a três registros Internet regionais e estas autoridades regionais são responsáveis por distribuir sub-blocos a ISPs dentro de suas regiões (veja a Tabela 13.6). Esses ISPs por sua vez dividem seus blocos e liberam trechos a clientes individuais. Somente grandes ISPs devem lidar diretamente com um dos registros de endereços patrocinados pelo ICANN.

A delegação do ICANN ao ARIN, RIPE e APNIC e depois para ISPs nacionais ou regionais permitiu uma maior agregação nas tabelas de roteamento do backbone. Clientes de ISP que receberam espaço de endereços dentro do bloco do ISP não precisam de entradas de roteamento individuais no backbone. Uma única entrada para o bloco agregado que aponta para o ISP já basta.

Tabela 13.6 Registros de endereços IP regionais.

Nome	Endereço Web	Região coberta
ARIN	www.arin.net	América do Norte e do Sul, África sub-Saara
APNIC	www.apnic.net	Região da Ásia/Pacífico
RIPE	www.ripe.net	Europa e adjacências

Originalmente, o espaço de endereços não era alocado de maneira muito razoável. O governo norte-americano reservou cerca de metade do espaço de endereços para ele e deu blocos relativamente pequenos para a Europa e Ásia. Porém a Europa e a Ásia gerenciaram seus espaços de endereço de modo muito mais inteligente do que os Estados Unidos. O mapa de espaço de endereços em

http://www.caida.org/outreach/resources/learn/ipv4space

ilustra esse fato de forma relativamente eficiente, mostrando o espaço de endereços IP como um todo, as porções que foram alocadas, as porções que foram roteadas (e, conseqüentemente, atingíveis) e os endereços para os quais o tráfego tem sido observado em alguns pontos de maior intercâmbio nos Estados Unidos.

Endereços privados e o NAT

Uma outra solução temporária para o esgotamento do espaço de endereços é o emprego de espaços de endereços IP privados, conforme descrito na RFC1918 (fevereiro de 1996). Na era do CIDR, os sites normalmente obtinham seus endereços IP de seus ISPs (Internet Service Providers, provedores de serviço Internet). Se um site quisesse mudar de ISP, isso poderia ser equivalente ao pagamento de um resgate, dado o alto custo para renumeração de suas redes. O ISP cedia o espaço de endereço enquanto ele permanecesse cliente. Se o site então quisesse optar por um outro ISP, ele teria de convencer o antigo ISP a permitir que continuasse com os endereços e também convencer o novo ISP a fazer com que o roteamento funcionasse corretamente para a nova localização com os antigos endereços. Tipicamente, os ISPs não querem esquentar a cabeça com esses tipos de problema e exigirão que o cliente renumere.

Uma alternativa para emprego de endereços atribuídos por ISPs é usar endereços privados que jamais são mostrados a seu ISP. A RFC1918 deixa de lado uma rede classe A, 16 redes classe B e 256 redes classe C que jamais serão alocadas globalmente e poderão ser usadas internamente por qualquer site. O ponto negativo é que jamais se deve permitir que pacotes pertencentes a esses endereços escapem para a Internet. Você deve filtrá-los em seu roteador de fronteira somente por garantia. Se alguns pacotes escaparem, você deve rastrear as configurações incorretas que possibilitaram que eles escapassem.

A Tabela 13.7 ilustra os números de rede reservados para endereçamento privado. (A coluna "Intervalo CIDR" mostra o intervalo para cada classe na notação CIDR mais compacta; ela não acrescenta nenhuma informação adicional.)

Os sites poderão optar dentre este conjunto pelo tamanho de rede que melhor atenda às necessidades de suas organizações.

14. Um novo registro está sendo criado para a América Latina, porém ele não estará operacional por um tempo.

Snow666

Tabela 13.7 Endereços IP reservados para uso privado.

Classe IP	De	Para	Intervalo CIDR
Classe A	10.0.0.0	10.255.255.255	10.0.0.0/8
Classe B	172.16.0.0	172.31.255.255	172.16.0.0/12
Classe C	192.168.0.0	192.168.255.255	192.168.0.0/16

Para permitir que hosts que usam endereços privados se comuniquem com a Internet, o roteador de fronteira do site executa um sistema denominado NAT (Network Address Translation, tradução de endereços de rede). O NAT intercepta pacotes endereçados com esses endereços de uso somente interno e reescrevem seus endereços de origem, usando um endereço IP externo real e talvez um número de porta de origem diferente. Ele também mantém uma tabela das associações (mapeamentos) que ele fez entre pares de endereço/porta de origem interno e externo para que a tradução possa ser executada ao contrário quando os pacotes de resposta chegarem da Internet.

O emprego de associação de número de porta por parte do NAT possibilita que várias conversações possam ser multiplexadas no mesmo endereço IP para que um único endereço externo possa ser compartilhado por vários hosts internos. Em alguns casos, um site pode dar conta com apenas um endereço IP "real".

Um site que usa o NAT ainda assim tem de solicitar espaço de endereços de seu ISP, porém a maioria dos endereços então obtidos é usada para mapeamentos NAT e não é atribuída a hosts individuais. Se, mais para a frente, o site quiser escolher um outro ISP, somente o roteador de borda e sua configuração NAT precisarão ser alterados e não as configurações dos hosts individuais.

Vários fornecedores de roteadores implementam o NAT, inclusive a Cisco. Também é possível ter um computador com Linux executando a função de NAT, embora muitos sites prefiram delegar esta tarefa a seus roteadores ou dispositivos de conexão de rede. Veja as seções específicas por fornecedor mais adiante neste capítulo para obter maiores detalhes. Por alguma razão, o mundo Linux costumava chamar o NAT de "mascaramento de IP". Entretanto, com a introdução do kernel 2.4 do Linux, o pessoal começou a chamá-lo de NAT também.

Uma configuração NAT incorreta pode permitir que pacotes do espaço de endereços privado escapem para a Internet. Os pacotes chegarão a seus destinos, porém pacotes de resposta não serão capazes de retornar. A CAIDA[15], uma organização que mede tudo a respeito de redes backbone, acredita que cerca de 0,1% a 0,2% dos pacotes no backbone possuam endereços privados ou checksums incorretos.

Isso parece ser uma porcentagem muito pequena, e realmente é, porém ela representa cerca de 20.000 pacotes a cada 10 minutos num circuito movimentadíssimo em MAE-West (uma das maiores centrais telefônicas públicas na qual vários ISPs se encontram para intercâmbio de tráfego). Vá ao site www.caida.org para obter outros dados estatísticos interessantes e ferramentas de medição de redes.

Uma desvantagem do NAT (ou quem sabe uma vantagem) é que um host arbitrário na Internet não pode se conectar diretamente às máquinas internas de sua instalação. Algumas implementações (por exemplo, Linux e Cisco PIX) permitem que configuremos "túneis" que suportam conexões diretas para determinados hosts.

Um outro problema é que alguns aplicativos embutiam endereços IP na porção dados dos pacotes; esses aplicativos são despistados ou confundidos pelo NAT. Entre os exemplos estão alguns protocolos de roteamento, programas de streaming como o RealVideo e SHOUTcast, alguns comandos FTP como PORT e PASV, o aplicativo de mensagens instantâneas ICQ e vários jogos. O NAT algumas vezes também interfere com VPNs (Virtual Private Networks, redes privadas virtuais).

Grandes corporações que adotam o NAT e endereços RFC1918 têm de instituir alguma maneira de coordenação centralizada de forma que todos os hosts, independentemente de seus grupos administrativos ou departamentais, tenham endereços IP únicos. A situação pode se tornar mais complicada quando uma empresa que usa espaço de endereços RFC1918 adquire ou se funde com uma outra que esteja fazendo a mesma coisa. Partes da organização combinada normalmente têm de ser renumeradas.

O NAT oculta a estrutura interna. Esse sigilo parece ser um ganho em termos de segurança, porém o pessoal especializado em segurança de sistemas diz que o NAT realmente não ajuda em nada em termos de segurança e certamente não substitui a necessidade de um firewall. Ele também frustra qualquer tentativa de medir o tamanho ou a topologia da Internet.

15. CAIDA (Cooperative Association for Internet Data Analysis, associação cooperativa para análise de dados Internet) sediada no San Diego Supercomputer Center no campus da UCSD (www.caida.org).

Snow666

Endereçamento IPv6

Um endereço IPv6 tem 128 bits de comprimento. Estes longos endereços destinavam-se originalmente a solucionar o problema da exaustão de endereços IP. Agora que estão aí, eles estão sendo explorados para ajudar em questões de roteamento, mobilidade e localidade de referência. Eles também se encontram (até o momento da publicação deste livro) num estado de desarranjo em relação aos registros DNS que serão usados para fazer a associação entre nomes de host e endereços IPv6.

Estão em uso dois tipos de registros DNS para endereço IPv6: o tipo A6, que é descrito a seguir e o tipo AAAA considerado anteriormente obsoleto, mas que agora está ressurgindo. Todo o mundo estava se virando com o AAAA até que o A6 pudesse ser empregado, porém agora o pessoal que participa das reuniões do IETF anda dizendo que o A6 é muito complicado e que devemos retornar para o AAAA. Portanto, esteja atento — pode ser que na época em que você estiver lendo este trecho, as próximas seções a seguir possam vir a ser apenas referências históricas de um padrão que não deu certo. Consulte o Capítulo 16 para obter maiores informações sobre o estado dos endereços IPv6 no DNS.

Os endereços IP jamais foram agrupados geograficamente da maneira como é feito com os números de telefone ou de códigos de endereçamento postal. Agora, com a segmentação proposta do espaço de endereços IPv6, eles pelo menos estarão agrupados junto aos ISPs. A fronteira entre a porção rede e a porção host de um endereço IPv6 é fixada em /64; o limite entre a topologia pública e a topologia local de um site é fixado em /48. A Tabela 13.8 ilustra as diversas partes de um endereço IPv6.

Tabela 13.8 As partes de um endereço IPv6.

Bits	Acrônimo	Tradução
1-3	FP	Prefixo do formato; o tipo de endereço, por exemplo, unidifusão
4-16	TLA ID	ID de agregação de alto nível, como o ISP backbone
17-24	RES	Reservado para uso futuro
25-48	NLA ID	ID de agregação do próximo nível, por exemplo, ISPs regionais e ID do site
49-64	SLA ID	ID de agregação em nível de site, como a sub-rede local
65-128	INTERFACE ID	Identificador de interface (endereço MAC mais preenchimento)

Dessas partes, somente o ID SLA e o ID INTERFACE pertencem ao host e seu site. As demais partes são fornecidas pelo ISP no sentido oposto do fluxo. O SLA especifica uma sub-rede local. O ID de interface de 64 bits identifica a interface de rede do host. Ele contém, tipicamente, o endereço MAC de 48 bits com os dígitos hexadecimais 0xFFFE no meio. Um bit especial no endereço MAC (bit 6 do primeiro byte, numerando os bits da esquerda para a direita, iniciando em 0) chamado de bit universal/local tem de ser complementado (veja a RFC2373). Esse esquema permite que os hosts sejam numerados automaticamente, o que é um recurso interessante para o administrador de sistemas, já que somente a sub-rede precisa ser gerenciada.

No IPv6, o endereço MAC é visto na camada IP, que tem tanto implicações boas como ruins. A marca e o modelo da placa de interface são codificados na primeira metade do endereço MAC, de modo que hackers de posse do código para uma determinada arquitetura serão beneficiados. A visibilidade dessas informações também preocupou alguns advogados especializados na área. O pessoal do IPv6 respondeu apontando que não era exigido o emprego de endereços MAC por parte dos sites; eles eram livres para usarem o que quisessem como endereço host. Um esquema para incluir um identificador aleatório na porção local do endereço também foi proposto. Muitos bits para se lidar!

Por outro lado, atribuir endereço IPv6 deveria ser mais fácil do que atribuir endereço IPv4, já que precisávamos apenas controlar o endereço de sub-rede. Os hosts são capazes de se autoconfigurarem — ou pelo menos, teoricamente.

Snow666

O prefixo de formato identifica o tipo de endereço IPv6: unidifusão, multidifusão ou qualquer tipo de difusão. Endereços unidifusão configuram FP em 001 (binário). Os IDs TLA e NLA identificam sua companhia telefônica de backbone IP de nível mais alto e os ISPs locais até a cadeia para o seu provedor de backbone.

A maioria dos fornecedores possui pilhas IPv6 em desenvolvimento ou já em uso. A Tabela 13.9 ilustra a disponibilidade do IPv6 por parte de alguns fornecedores de roteadores e para alguns sistemas operacionais comuns. (Os comutadores (switches) na verdade não se importam com o IPv6, pois eles não tomam decisões de roteamento e não analisam o interior do cabeçalho IP.)

Para uma visão completa das implementações do IPv6 no Linux, visite os seguintes sites:

playground.sun.com/pub/ipng/html/ipng-implementations.html
www.bieringer.de/linux/IPv6/status/IPv6+Linux-status-distributions.html

O último site descreve cinco implementações para o Linux e exibe uma interessante tabela que inclui não somente as implementações IPv6 básicas, como também diversos utilitários de rede comuns.

Tabela 13.9 Disponibilidade do IPv6 por parte de alguns fornecedores.

Sistema	IPv6?	Comentários
Red Hat	sim	IPv6 no kernel do Linux, versões 2.2 e posteriores
SuSE	sim	Inclui uma versão IPv6 dos protocolos de roteamento OSPF e RIP e vários utilitários
Debian	sim	Menos utilitários na 2.2, melhor na 3.0, porém ainda faltando **inetd**
Windows XP	sim	Disponível versão de pesquisa (incluindo código-fonte)
Cisco	sim	Implementado de forma lenta (CPU)[a]
Juniper	sim	Implementado de forma rápida
Nortel/Bay	sim	Embarcado desde 1997

a. Provavelmente mudará para a forma rápida caso os clientes exijam.

Das distribuições Linux usadas como exemplo neste livro, o SuSE parece ser o mais completo na área do IPv6, seguido pelo Red Hat e com o Debian carregando a lanterna. De acordo com esse site, a implementação IPv6 Linux mais completa é a distribuição Linux polonesa no site www.ipd.org.pl.

Os órgãos responsáveis pelos registros de endereços acabaram de começar a alocar espaço de endereços IPv6. No momento, o ARIN somente alocará espaço para grandes ISPs que planejem implementar uma rede IPv6 dentro dos próximos 12 meses. Esses ISPs podem então alocar subespaços a seus clientes. Muito embora tenhamos 665.579.793.348.866.943.898.599 endereços IPv6 por metro quadrado da superfície terrestre (realmente!), a alocação parece parcimoniosa e é cara. A taxa mínima é de US$ 2.500/ano e a máxima de US$ 20.000/ano; assinantes IPv4 parecem estar isentos.

Eis algumas fontes úteis sobre IPv6 (todas em inglês)

- www.6bone.net — o banco de teste para IPv6; usa túneis para evitar roteadores IPv4
- www.6ren.net — uma rede educativa e de pesquisa mundial sobre IPv6
- www.ipv6.org — FAQs e informações técnicas
- www.ipv6forum.com — para o pessoal de marketing e propaganda do IPv6

Uma grande vantagem do IPv6 é que ele foi desenvolvido para solucionar a questão da renumeração.[16] No mundo IPv4, os ISPs alocam espaço de endereços para os clientes, porém os endereços não são portáveis; quando os clientes abandonam um ISP, eles têm de devolver seus endereços e fazer a renumeração de acordo com endereços fornecidos pelo novo ISP. No IPv6, o novo ISP lhe dá um prefixo de endereço que você simplesmente anexa às porções locais de seus endereços, provavelmente no seu roteador de fronteira. Esse esquema é similar àquele do NAT para endereço IPv4, mas sem qualquer um dos pequenos problemas do NAT.

16. O IETF está pendente entre os esquemas AAAA e A6 para registros de endereços DNS para hosts IPv6. A6 é mais complicado, mas soluciona a questão da renumeração. O AAAA é mais simples e poderia ser empregado mais cedo, porém ele esbarra no problema da renumeração. A RFC 3363, de agosto de 2002, considera o problema da numeração do IPv6 e faz a seguinte recomendação: manter a RFC 1886 no "standards track" (trilha dos padrões) e que seja avançada ao próximo nível, enquanto se move a RFC 2874 a um status Experimental. Ou seja, ainda não há uma definição com relação a numeração IPv6 e registros DNS.

www.snow666.blogspot.com
Snow666

Foram propostos diversos esquemas para facilitar a transição do sistema IPv4 para o IPv6, inclusive o emprego do NAT para ocultar endereços IPv6 enquanto pacotes são tunelados através da infra-estrutura IPv4 existente.

13.5 ROTEAMENTO

Roteamento é o processo de direcionar um pacote através do emaranhado de redes que se interpõe entre a sua origem e o seu destino. No sistema TCP/IP, ele é similar a pedir informações num país onde não estejamos familiarizados. A primeira pessoa com que você conversa poderia lhe indicar a cidade correta. Assim que você estivesse um pouco mais próximo de seu destino, a próxima pessoa poderia lhe informar como chegar à rua correta. Eventualmente, você chegaria perto o bastante para que uma outra pessoa fosse capaz de identificar o prédio que você está procurando.

As informações de roteamento TCP/IP assumem a forma de regras ("rotas") como, por exemplo, "Para atingir a rede A, envie pacotes através da máquina C". Também pode haver uma rota-padrão que informa o que fazer com pacotes destinados a uma rede para a qual não há rota explícita.

As informações de roteamento são armazenadas numa tabela no kernel. Cada entrada de tabela possui vários parâmetros, entre os quais uma máscara de rede para cada rede listada (outrora opcional, porém agora exigida se a máscara de rede padrão não for correta). Para encaminhar um pacote para um determinado endereço, o kernel escolhe a rota mais específica daquelas que atendem aos critérios especificados (isto é, aquela com a máscara de rede mais longa). Se o kernel não encontrar nenhuma rota relevante e nenhuma rota-padrão, então ele retorna um erro ICMP "network unreachable" (rede inatingível) para o emissor.

A palavra "roteamento" é usada comumente com dois significados:

- Pesquisar um endereço de rede na tabela de roteamento para encaminhar um pacote em direção a seu destino.
- Construir a tabela de roteamento primária.

Nesta seção examinaremos a função de encaminhamento e veremos como as rotas podem ser adicionadas ou eliminadas manualmente da tabela de roteamento. Postergamos o tópico mais complicado de protocolos de roteamento que constroem e mantêm a tabela de roteamento para o Capítulo 14.

Tabelas de roteamento

Podemos examinar a tabela de roteamento de uma máquina por meio do comando **netstat -r**. Use **netstat -rn** para evitar pesquisas DNS e para apresentar todas as informações na forma numérica. Discutiremos o comando **netstat** de maneira mais detalhada a partir da página 473, porém eis um pequeno exemplo para lhe dar uma idéia melhor sobre o aspecto das rotas. Esse host possui duas interfaces de rede: 132.236.227.93 (eth0) na rede 132.236.227.0/24 e 132.236.212.1 (eth1) na rede 132.236.212.0/26.

```
% netstat -rn
Kernel IP routing table
Destination     Genmask         Gateway         Fl  MSS   Iface
132.236.227.0   255.255.255.0   132.236.227.93  U   1500  eth0
default         0.0.0.0         132.236.227.1   UG  1500  eth0
132.236.212.0   255.255.255.192 132.236.212.1   U   1500  eth1
132.236.220.64  255.255.255.192 132.236.212.6   UG  1500  eth1
127.0.0.1       255.255.255.255 127.0.0.1       U   3584  lo
```

O campo de destino normalmente é um endereço de rede; o gateway tem de ser um endereço de host. Por exemplo, a quarta rota diz que para alcançar a rede 132.236.220.64/26, os pacotes têm de ser enviados para o gateway 132.236.212.6 por meio da interface eth1. A segunda entrada é a rota-padrão; pacotes não endereçados explicitamente a qualquer uma das três redes listadas (ou para a máquina em si) serão enviados para o host gateway padrão, 132.236.227.1. Os hosts são capazes de rotear somente para máquinas gateway que estão conectadas diretamente às mesmas redes.

*Consulte a seção 13.7 para obter maiores informações sobre o comando **route**.*

As tabelas de roteamento podem ser configuradas estaticamente, dinamicamente ou com uma combinação dos dois métodos. Uma rota estática é uma que introduzimos explicitamente por meio do comando **route**. Rotas estáticas devem permanecer na tabela de roteamento enquanto o sistema estiver no ar; normalmente elas são configuradas no momento do boot por meio de um dos scripts de inicialização do sistema. Por exemplo, os comandos Linux a seguir

```
route add -net 132.236.220.64 netmask 255.255.255.192 gw 132.236.212.6 eth1
route add default gw 132.236.227.1 eth0
```

www.snow666.blogspot.com
Snow666

acrescentariam a quarta e a segunda rotas exibidas pelo comando **netstat -rn** acima. (A primeira e a terceira rotas nessa tela foram adicionadas por **ifconfig** quando as interfaces eth0 e eth1 foram configuradas.)

A rota final também é acrescentada no momento do boot. Ela configura um pseudodispositivo chamado interface loopback. A interface loopback impede que pacotes enviados do host para si mesmo saiam da rede; em vez disso, eles são transferidos diretamente da fila de saída da rede para a fila de entrada da rede dentro do kernel.

Em uma rede local estável, o roteamento estático é uma solução eficiente. Ele é fácil de ser gerenciado e é confiável. Porém, requer que o administrador de sistemas conheça minuciosamente a topologia da rede no momento do boot e que a topologia não mude com freqüência.

A maioria das máquinas numa rede local possui apenas um caminho para atingir do restante da rede e, portanto, o problema do roteamento é simples. Uma rota-padrão adicionada no momento do boot já é suficiente para apontar o caminho a seguir. Hosts que usam DHCP (veja a seção 13.9) para obter seus endereços IP também podem obter uma rota-padrão por meio de DHCP.

Para topologias de rede mais complicadas é necessário o emprego de roteamento dinâmico. O roteamento dinâmico é executado tipicamente por um processo daemon que mantém e modifica a tabela de roteamento. Os daemons de roteamento em hosts distintos se comunicam para descobrir a topologia da rede e para descobrir como atingir destinos distantes. Há disponíveis vários daemons de roteamento. No Capítulo 14, descrevemos o daemon Linux padrão, **routed**, e um daemon mais incrementado chamado **gated**, bem como os protocolos de roteamento que eles "falam".

Redirecionamentos ICMP

Embora o IP geralmente não se preocupe propriamente com o gerenciamento das informações de roteamento, ele define efetivamente um pequeno recurso de controle de danos chamado de redirecionamento ICMP. Quando um roteador encaminha um pacote a uma máquina na mesma rede da qual o pacote foi originalmente recebido, certamente algo está errado. Já que o remetente, o roteador, e o roteador do próximo salto se encontram todos na mesma rede, o pacote poderia ter sido encaminhado em um salto em vez de dois. O roteador pode concluir que as tabelas de roteamento do remetente são imprecisas ou incompletas.

Nessa situação, o roteador pode notificar o emissor de seu problema com o pacote de redirecionamento ICMP. Basicamente, um redirecionamento diz: "Você não deve ficar enviando pacotes para o host *xxx* para mim, mas sim para o host *yyy*". O protocolo ICMP permite que sejam enviados redirecionamentos tanto para endereços de host individuais como para redes inteiras. Porém, muitas implementações geram somente redirecionamentos para hosts; os redirecionamentos para redes são muito menos úteis nos dias de hoje, pois eles se aplicam somente para redes classes A, B ou C.

Após receber um redirecionamento, um emissor gentil atualiza sua tabela de roteamento de modo que pacotes futuros associados àquele destino adotarão o caminho mais direto. Nos primórdios da multidifusão, alguns poucos sistemas geravam redirecionamentos de roteamento ICMP em resposta a pacotes multidifusão. Os sistemas modernos não apresentam esse problema.

A situação ICMP padrão não contém nenhuma etapa de autenticação. Seu roteador recebe um redirecionamento que se diz ser pertencente a outro roteador bem respeitado e o direciona para enviar tráfego para algum outro lugar. Você deve dar ouvidos a ele? Dar atenção a redirecionamentos na verdade cria um certo tipo de problema de segurança. Os redirecionamentos geralmente são ignorados pelo Linux (por questões de segurança) e por roteadores Cisco (pois eles são roteadores). Não é uma boa idéia permitir que hosts não confiáveis modifiquem suas tabelas de roteamento.

No Linux, a variável **accept_redirects** na hierarquia **/proc** controla a aceitação de redirecionamentos ICMP. Consulte a seção 13.10 para obter instruções sobre como examinar e reinicializar essa variável.

13.6 ARP: O Protocolo de Determinação de Endereços

Muito embora os pacotes IP normalmente sejam interpretados em termos de endereços IP, os endereços de hardware têm de ser usados para transportarem efetivamente dados através da camada de link de uma rede.[17] O ARP (Address Resolution Protocol, protocolo de determinação de endereços) descobre o endereço de hardware associado a um determinado endereço IP. Ele pode ser usado em qualquer tipo de rede que suporte difusão ampla mas é descrito comumente em termos de redes Ethernet.

Se o host A quer enviar um pacote para o host B na mesma rede Ethernet, ele usa o ARP para descobrir o endereço de hardware de B. Se B não se encontrar na mesma rede que A, o host A usa o sistema de roteamento para de-

17. Exceto em links ponto-a-ponto, no qual a identidade do destino algumas vezes já está implícita.

Snow666

terminar o roteador do próximo salto juntamente com a rota para B e depois usa o ARP para descobrir o endereço de hardware deste roteador. Já que o ARP usa pacotes de transmissão ampla, que não são capazes de cruzar redes[18], ele pode ser usado somente para descobrir os endereços de hardware das máquinas conectadas diretamente à rede local do host emissor.

Toda máquina mantém uma tabela na memória chamada cache ARP contendo os resultados de consultas ARP recentes. Sob circunstâncias normais, muitos dos endereços que um host necessita são descobertos logo após a inicialização, de modo que o ARP não é importante por grande parte do tráfego de rede.

O ARP funciona enviando por difusão ampla[19] um pacote do tipo: "Alguém por acaso conhece o endereço de hardware para 128.138.116.4?". A máquina que está sendo procurada reconhece o seu próprio endereço IP e retorna uma resposta: "Sim, este é o endereço IP atribuído a uma das minhas interfaces de rede e o endereço Ethernet correspondente é 8:0:20:0:fb:6a".

A consulta original inclui os endereços IP e Ethernet do solicitante para que a máquina que está sendo procurada possa responder sem enviar uma consulta ARP própria. Conseqüentemente, as duas máquinas tomam conhecimento das associações ARP de cada uma delas usando somente um intercâmbio de pacotes. Outras máquinas que "grampeiam" a difusão ampla do solicitante também são capazes de registrar o seu mapeamento de endereços. Essa inspeção passiva de tráfego ARP é chamada algumas vezes de bisbilhotice.

O comando **arp** examina e manipula o cache ARP do kernel. **arp** é usado tipicamente para adicionar ou eliminar uma entrada; ele também é capaz de descarregar a tabela ou exibi-la. O comando **arp -a** exibe o conteúdo do cache ARP. Por exemplo:

```
redhat% /sbin/arp -a
sprint-gw (192.168.1.254) at 00:02:4B:5B:26:45 [ether] on eth0
inura-local.toadranch.com (192.168.1.101) at 00:04:76:37:AE:7E [ether] on eth0
```

O comando **arp** é geralmente útil somente para depuração e para situações que envolvam hardware especial. Alguns dispositivos não são espertos o bastante para compreenderem ARP (por exemplo, impressoras conectadas à rede ou monitores gráficos de propósito especial). Para suportar tais dispositivos, talvez seja necessário configurar uma outra máquina como servidor ARP proxy para o seu hardware incapaz. Isso normalmente também é feito com o comando **arp** (usando o flag **-s**). Por exemplo:

```
# /sbin/arp -s printer.toadranch.com 00:50:04:ce:ef:38
```

Os kernels Linux versão 2.4 e posteriores não suportam serviço ARP proxy para uma sub-rede inteira, porém atuarão automaticamente como um servidor ARP proxy quando existir uma rota e a interface for configurada para encaminhar pacotes.

Se dois hosts numa rede estão usando o mesmo endereço IP, um deles terá a entrada de tabela ARP correta e o outro ficará com a errada. Podemos usar o comando **arp** para rastrear a máquina problemática.

Algumas vezes, endereços de hardware precisam ser convertidos em endereços IP. Uma série de equipamentos deficientes (por exemplo, estações de trabalho sem disco, computadores de rede, impressoras) precisa fazer essa conversão no momento do boot. Em vez de ter um endereço IP definido diretamente num arquivo de configuração, uma máquina pode consultar um servidor central para descobrir o seu próprio endereço. O protocolo quase obsoleto RARP (Reverse ARP) estende o ARP para abranger conversões reversas.

Diferentemente do ARP, o RARP requer que um processo de servidor centralizado seja instalado em cada rede. O RARP não é autoconfigurável; temos de fornecer um mapeamento explícito entre endereços Ethernet e endereços IP. Na maioria dos sistemas que suportam RARP, o servidor é chamado **rarpd** e dados de configuração são extraídos de **/etc/ethers** e **/etc/hosts**.

O RARP foi suplantado em muito, primeiramente pelo BOOTP e agora pelo DHCP.

Isso encerra nossa cobertura sobre fundamentos de rede. Nas seções a seguir, tratamos das questões envolvidas na configuração de máquinas Linux para sua máquina local e Internet.

13.7 Adicionando uma Máquina a uma Rede

Somente alguns poucos passos estão envolvidos na adição de uma máquina nova a uma rede local existente, porém alguns fornecedores ocultam os arquivos que você tem de modificar e geralmente dificultam esta tarefa. Outros fornecem um script de configuração que solicita parâmetros de rede necessários, que é bom até o momento em que

18. Os roteadores normalmente podem ser configurados para inundar pacotes de difusão ampla em outras redes; não faça isso.
19. O ARP usa a camada subjacente das convenções de difusão ampla da camada de link, não a difusão ampla IP.

www.snow666.blogspot.com
Snow666

você precisar desfazer algo ou transferir uma máquina. Antes de ativar uma máquina nova numa rede que está conectada à Internet, você deve executar procedimentos de segurança nesta máquina (Capítulo 21, *Segurança*) para que você não convide inadvertidamente hackers para sua rede local.

As etapas básicas para se adicionar uma nova máquina a uma rede local são:

- Atribuir um nome de host e um endereço IP exclusivos.
- Definir o novo host para configurar suas interfaces de rede no momento do boot.
- Configurar uma rota-padrão e talvez roteamento mais extravagante.
- Apontar para um servidor de nomes DNS, para possibilitar o acesso ao restante da Internet.

Consulte também o Capítulo 20, Gerenciamento de Redes e Depuração.

Obviamente, você também poderia acrescentar uma etapa de depuração a essa seqüência. Após qualquer modificação que possa vir a afetar a inicialização, você sempre deve reiniciar para verificar se a máquina é acionada corretamente. Seis meses depois quando a alimentação falhar e a máquina se recusar a inicializar, será difícil se lembrar de que alteração você fez que poderia estar causando o problema.

Se sua rede usa DHCP (Dynamic Host Configuration Protocol, protocolo de configuração de host dinâmico) o servidor DHCP fará essas tarefas para você. Veja a seção 13.9, sobre DHCP para obter informações gerais sobre o DHCP e os detalhes de como configurar as distribuições adotadas neste livro para usar o DHCP no momento da inicialização.

Veremos primeiramente uma descrição genérica dessas etapas, depois retornaremos para os detalhes de cada distribuição numa série de seções específicas conforme o fornecedor. Os comandos envolvem a camada de rede do kernel Linux e são os mesmos em várias distribuições diferentes. Porém, cada distribuição estabeleceu seus próprios arquivos de configuração para configuração de rede automática no momento da inicialização, conforme sintetizado na Tabela 13.10.

Tabela 13.10 Arquivos de configuração de rede.

Sistema	Arquivo	O que é configurado aí
Red Hat	**/etc/sysconfig/network** **network-scripts/ifcfg-** *ifname*	Nome do host, endereço IP de rota padrão, máscara de rede, endereço de difusão ampla
SuSE	**/etc/rc.config** **/etc/route.conf**	Nome do host, endereço IP, máscara de rede e outros Rota-padrão
Debian	**/etc/hostname** **/etc/network/interfaces**	Nome do host Endereço IP, máscara de rede, rota-padrão

O processo de projetar e instalar uma rede física é visto no Capítulo 15, *Hardware de Rede*. Caso esteja lidando com uma rede existente e tenha uma noção geral de como ela está configurada, talvez não seja necessário você ler muito sobre os aspectos físicos de redes a menos que planeje estender a rede existente.

Descrevemos o processo de configuração de rede em termos de redes Ethernet; outras tecnologias são, basicamente, similares.

Atribuindo nomes de host e endereços IP

Os administradores têm várias teorias sobre como a associação entre nomes de hosts e endereços IP é melhor mantida numa instalação local: por meio do arquivo **hosts** file, do NIS, do sistema DNS ou, quem sabe, alguma combinação destas. Se forem usados vários sistemas, também deve haver um planejamento sensato de como eles funcionarão juntos. Os valores conflitantes são escalabilidade e manutenção *versus* um sistema que seja suficientemente flexível para permitir que máquinas inicializem quando nem todos os serviços estarão disponíveis (e suficientemente flexíveis para lidar com a heterogeneidade de sua rede).

Veja o Capítulo 16 para maiores informações sobre DNS.

Uma outra questão a longo prazo que pode ser relevante é aquela da renumeração. Se sua instalação muda de ISP, talvez você precise devolver os seus antigos endereços IP e fazer uma renumeração de acordo com endereços atribuídos pelo novo ISP. Esse processo torna-se bastante desalentador caso você tenha de visitar cada host da rede para reconfigurá-lo. Para agilizar tal processo de renumeração, use nomes de host nos arquivos de configuração e faça a conversão nome de host-endereço IP somente nos arquivos de bancos de dados DNS. Por outro lado, usar endere-

www.snow666.blogspot.com
Snow666

ços IP nos arquivos de configuração reduz as dependências durante a inicialização quando nem todos os serviços estarão disponíveis. Maldição se você faz, maldição se você não faz.

O arquivo **/etc/hosts** é a maneira mais antiga e simples de associar nomes a endereços IP. Cada linha começa com um endereço IP e prossegue com os diversos nomes simbólicos pelos quais este endereço é conhecido. Eis um arquivo **/etc/hosts** típico para o host lollipop:

```
127.0.0.1       localhost
192.108.21.48   lollipop.xor.com lollipop loghost
192.108.21.254  chimchim-gw.xor.com chimchim-gw
192.108.21.1    ns.xor.com ns
192.225.33.5    licenses.xor.com license-server
```

Uma versão mínima conteria somente as duas primeiras linhas. É comum ter localhost como a primeira entrada no arquivo **/etc/hosts**.

Pelo fato de **/etc/hosts** conter somente associações locais, a maioria dos sistemas modernos o usa somente para associações que são necessárias no momento da inicialização. O DNS é então consultado para localizar associações para o restante da rede local e para todos os demais. Algumas vezes **/etc/hosts** é usado para especificar associações que não desejamos que o restante das pessoas conheçam e, portanto, não publicar no DNS.

/etc/hosts já foi importante durante o processo de inicialização pois o DNS ainda não se encontrava disponível e os nomes de host eram algumas vezes usados em arquivos de configuração em vez de endereços IP. As distribuições Linux modernas realmente não precisam do arquivo **/etc/hosts**, mas elas devem provavelmente conter ao menos as associações para o próprio host e o endereço de loopback. Associações para a máquina gateway padrão e um servidor de nomes provavelmente também seriam úteis. Muitas instalações colocam todos os seus hosts, servidor e gateways realmente importantes no arquivo **/etc/hosts**. Outros colocam somente o host em si e a interface loopback; outros ainda colocam todos os hosts locais e seus servidores de nomes de backup fora da instalação.

Os sistemas Linux aqui apresentados instalam somente um arquivo **/etc/hosts** muito básico. O arquivo do Debian contém apenas o host local, o do Red Hat contém o host local e a máquina em si e o do SuSE contém o host local, a própria máquina e endereços IPv6 para o host local e alguns poucos nomes IPv6 especiais.

Se seu arquivo **/etc/hosts** contiver todos os seus dados locais, ele deve ser replicado em cada uma das máquinas que quiser usar nomes simbólicos. Vários esquemas permitem que uma única versão do arquivo **hosts** seja mantida numa localização centralizada e seja distribuída ou compartilhada por outros hosts em sua instalação; veja o Capítulo 18, *Compartilhando Arquivos de Sistema*, para obter maiores informações. DNS é realmente a maneira "correta" de gerenciarmos o mapeamento. O Capítulo 16 descreve detalhadamente o DNS e o BIND, sua implementação Linux/Unix.

Veja o Capítulo 18 para obter maiores informações sobre o LDAP.

O comando **hostname** atribui um nome de host a uma máquina. **hostname** é normalmente executado no momento do boot por meio de um dos scripts de inicialização, que obtém o nome a ser atribuído por meio do arquivo de configuração. Obviamente, cada fornecedor dá nomes diferentes a esse arquivo de configuração. Veja as seções específicas a cada fornecedor, a partir da seção 13.8, para obter maiores informações sobre sua distribuição específica. A maioria dos sistemas hoje em dia atribui um nome totalmente qualificado (isto é, um nome que inclui tanto o nome do host quanto o nome de domínio DNS como, por exemplo, anchor.cs.colorado.edu).

Numa instalação pequena, você pode distribuir facilmente nomes de hosts e endereços IP manualmente. Porém, quando várias redes e vários grupos administrativos distintos estão envolvidos, ajuda ter alguma espécie de coordenação central para garantir exclusividade. Para parâmetros de rede atribuídos dinamicamente, o DHCP cuida das questões de exclusividade. Algumas instalações agora estão usando bancos de dados LDAP para controlarem suas atribuições de nomes de hosts e endereços IP.

ifconfig: configura interfaces de rede

ifconfig habilita ou desabilita uma interface de rede, configura o seu endereço IP e máscara de sub-rede e configura várias outras opções e parâmetros. Ele é normalmente executado no momento do boot (com parâmetros de linha de comando extraídos de arquivos de configuração), mas também pode fazer alterações com o sistema em andamento. Tome cuidado se estiver fazendo alterações **ifconfig** e estiver logado remotamente; muitos administradores de sistemas ficaram bloqueados desta maneira e tiveram que se dirigir até o local para corrigir o problema.

Um comando **ifconfig** mais comumente assume a seguinte forma:

```
ifconfig interface address options …
```

www.snow666.blogspot.com
Snow666

Por exemplo:

```
ifconfig eth0 128.138.240.1 netmask 255.255.255.0 up
```

interface identifica a interface de hardware a qual o comando se aplica. Em sistemas UNIX este normalmente é um nome de dispositivo de dois a três caracteres (derivados do chipset usado na placa de rede) seguida de um número, porém para o Linux quase sempre é algo do tipo eth0.[20] A identidade real do hardware e a associação a um driver de dispositivo apropriado são guardados no arquivo **/etc/modules.conf** numa linha de nome alternativo. A interface loopback chama-se lo.

ifconfig *interface* exibe as configurações atuais para *interface* sem alterá-las. Muitos sistemas interpretam **-a** como significando "all interfaces", ou seja, todas as interfaces, e **ifconfig -a** pode, conseqüentemente, ser usado para descobrir que interfaces estão presentes no sistema. Se o seu sistema não compreende **ifconfig -a**, tente **netstat -i** para descobrir os nomes das interfaces.

O parâmetro *address* especifica o endereço IP da interface. Muitas versões de **ifconfig** também aceitam um nome de host para o parâmetro de endereço. Preferimos usar o endereço IP real; se **ifconfig** receber um nome de host (ou a saída do comando **hostname**), o potencial para surgimento de problemas na hora da inicialização aumenta. Se houver um problema na determinação do nome de host, a máquina não inicializará ou inicializará num estado no qual não possa ser acessada da rede, necessitando que você vá em pessoa até a máquina para depurar o problema. Consultas DNS que não são capazes de serem completadas levam um bom tempo para atingir o tempo de time out, fazendo parecer que a máquina está suspensa. Por outro lado, se alguma vez você tiver de renumerar a sua rede, localizar todos esses endereços IP definidos diretamente e ocultos nos arquivos de configuração pode se tornar um pesadelo.

A palavra-chave **up** ativa a interface; **down** a desativa. Quando um comando **ifconfig** atribui um endereço IP a uma interface, como no exemplo acima, o parâmetro **up** está implícito e não precisa ser mencionado pelo nome.

ifconfig reconhece muitas outras opções. Veremos somente as mais comuns; como sempre, consulte suas páginas de manual para a palavra final sobre o seu sistema em particular. Todas as opções de **ifconfig** possuem nomes simbólicos. Algumas opções exigem um argumento, que poderia ser colocado imediatamente após o nome da opção.

A opção **netmask** configura a máscara de sub-rede para a interface e é necessária caso a rede não esteja dividida em sub-redes de acordo com sua classe de endereços (A, B ou C). A máscara pode ser especificada na notação ponto decimal ou na forma de um número hexadecimal de quatro bytes começando com **0x**. Em qualquer um dos casos, bits configurados em 1 fazem parte do número da rede e bits configurados em 0 fazem parte do número do host.

A opção **broadcast** especifica o endereço IP de difusão ampla para a interface, expresso tanto em notação hexadecimal como pontual. O endereço de difusão ampla correto é um no qual a porção host esteja configurada toda em 1, e a maioria dos sistemas usa como padrão este valor; eles utilizam a máscara de rede e o endereço IP para calcular o endereço de difusão ampla.

No Linux, podemos configurar o endereço de difusão ampla para qualquer endereço IP que seja válido para a rede na qual o host está conectado. Alguns sites optaram por valores estranhos como endereços de difusão ampla na esperança de evitarem determinados tipos de ataques de rejeição de serviços que se baseiam em pings de difusão ampla. Não gostamos dessa abordagem por diversas razões.

Primeiramente, ela requer que você reinicialize o endereço de difusão ampla em cada host da rede local, o que pode tomar muito tempo numa rede extensa. Em segundo lugar, requer que você esteja absolutamente certo de que reconfigurou todos os hosts, caso contrário, isto poderá resultar em tempestades de difusão ampla nas quais os pacotes trafegam de máquina em máquina até que seus TTLs expirem.

As tempestades de difusão ampla ocorrem porque o mesmo endereço de difusão ampla da camada de link tem de ser usado para transportar pacotes independentemente de como o endereço IP de difusão ampla tenha sido configurado. Suponha, por exemplo, que a máquina X imagine que o endereço de difusão ampla seja A1 e que a máquina Y imagine que ele seja A2. Se X enviar um pacote para o endereço A1, Y receberá o pacote (pois o endereço de destino da camada de link é o endereço de difusão ampla), verá que o pacote não é para si mesmo e nem para o endereço de difusão ampla (pois Y pensa que o endereço de difusão ampla é A2) e então encaminhará[21] o pacote de volta

20. Podemos atribuir mais de um endereço IP a uma interface fazendo uso do conceito de interfaces de rede virtuais ou de nomes alternativos IP. Os administradores de sistemas fazem isso com freqüência para permitirem que uma máquina hospede vários sites Web. Em sistemas Linux, as interfaces virtuais são denominadas eth0:0, eth0:1 e assim por diante. Não precisamos declarar as interfaces com antecedência, apenas aplicar o comando **ifconfig** a elas para configurá-las. Veja o Capítulo 22 para obter maiores informações.
21. A máquina Y tem de ser configurada com `ip_forwarding` ativado para que isso aconteça.

www.snow666.blogspot.com
Snow666

para a rede. Se duas máquinas estiverem no estado de Y, o pacote circulará até que o seu TTL expire. Tempestades de difusão ampla podem erodir sua largura de banda, especialmente numa grande rede comutada.

Uma forma melhor de evitar problemas com pings de difusão ampla é impedir que seus roteadores de fronteira os encaminhem e informar hosts individuais para não responderem a eles. Veja a seção 13.10 para obter instruções de como implementar essas restrições.

No exemplo de **ifconfig** no início desta seção, o endereço de difusão ampla é 128.138.240.255, pois a rede é do tipo /24, conforme especificado pelo valor de máscara de rede de 255.255.255.0.

Executar **ifconfig** gera o seguinte:

```
redhat% /sbin/ifconfig eth0
eth0  Link encap:Ethernet HWaddr 00:02:B3:19:C8:86
      inet addr:192.168.1.13 Bcast:192.168.1.255 Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:206983 errors:0 dropped:0 overruns:0 frame:0
      TX packets:218292 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      Interrupt:7 Base address:0xef00
```

A falta de colisões na interface Ethernet deste exemplo pode indicar uma rede pouco carregada ou, mais provavelmente, uma rede comutada. Numa rede compartilhada (construída com hubs em vez de switches), devemos verificar esse número para garantir que ele esteja abaixo de cerca de 5% dos pacotes de saída. Um grande número de colisões indica uma rede carregada que precisa ser observada e possivelmente dividida em várias sub-redes ou então migrada para uma infra-estrutura comutada.

Vejamos alguns exemplos completos.

```
# ifconfig lo 127.0.0.1 up
```

Esse comando configura a interface de loopback, que normalmente não requer que seja configurada nenhuma opção. Você jamais deve alterar a configuração-padrão de seu sistema para essa interface. A máscara de rede implícita de 255.0.0.0 é correta e não precisa ser sobregravada manualmente.

```
# ifconfig eth0 128.138.243.151 netmask 255.255.255.192
   broadcast 128.138.243.191 up
```

Esse é um exemplo típico para uma interface Ethernet. Os endereços IP e de difusão ampla são configurados, respectivamente, para 128.138.243.151 e 128.138.243.191. A rede é classe B (isto pode ser visto através do primeiro byte do endereço), porém ela foi dividida em sub-redes por 10 bits adicionais numa rede /26. O 192 na máscara de rede é 11000000 em binário e, portanto, adiciona 2 bits extras aos 24 contidos nos três octetos 255. O 191 no endereço de difusão ampla é 10111111 em binário, que configura todos os 6 bits de host em 1 e indica que esta interface faz parte da terceira rede (os primeiros dois bits são 10) no grupo de 4 extraídos do 4º octeto.

Agora que você já sabe como configurar manualmente uma interface de rede, precisa descobrir como os parâmetros de **ifconfig** são configurados quando a máquina é inicializada e precisa certificar-se de que os novos valores sejam introduzidos corretamente. Normalmente fazemos isso editando um ou mais arquivos de configuração; veja as seções específicas a cada fornecedor começando na página 206 para obter maiores informações.

mii-tool: configura autonegociação e outras opções específicas da mídia

Ocasionalmente, o hardware de rede tem opções configuráveis que são específicas ao seu tipo de mídia. Um exemplo extremamente comum disso é a Ethernet dos dias atuais, em que uma placa pode suportar 10, 100 ou até mesmo 1000 Mb/s tanto em half-duplex quanto em full-duplex. A maioria dos equipamentos tem como padrão o modo de autonegociação, no qual tanto a placa como a conexão acima (normalmente uma porta de um switch) tentam adivinhar o que o outro deseja usar. Infelizmente, o processo de autonegociação normalmente funciona tão bem quanto tentar laçar um novilho quando se está com os olhos vendados. Altas taxas de perdas de pacotes (especialmente de pacotes grandes) normalmente é resultado de autonegociação que falhou.

A melhor maneira de evitar essa armadilha é fixar a velocidade da interface e o modo duplex tanto nos servidores quanto nas portas de comutação nas quais eles estão conectados. A autonegociação é útil para portas em áreas públicas em que laptops de usuários em percurso podem dar uma parada para uma visita, mas não é de nenhuma utilidade para hosts conectados estaticamente. Se estiver tendo problemas com perdas misteriosas de pacotes, desative a autonegociação em todos os pontos como primeira medida.

www.snow666.blogspot.com
Snow666

No Linux, o comando **mii-tool** consulta e configura parâmetros específicos de mídia como velocidade do link e modo duplex. Podemos verificar o estado de uma interface através do flag **-v**. Por exemplo, a interface eth0 a seguir tem a autonegociação habilitada:

```
% mii-tool -v eth0
eth0: negotiated 100baseTx-FD flow-control, link ok
   product info: vendor 00:10:5a, model 0 rev 0
   basic mode:   autonegotiation enabled
   basic status: autonegotiation complete, link ok
   capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
   Advertising:  100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
   link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
```

Para fixar essa interface em 100 Mb/s modo full-duplex, use o comando

```
# mii-tool -force=100BaseTx-FD eth0
```

Após isso, a consulta de estado retorna

```
% mii-tool -v eth0
eth0: 100 Mbit, full duplex, link ok
   product info:   vendor 00:10:5a, model 0 rev 0
   basic mode:     100 Mbit, full duplex
   basic status:   link ok
   capabilities:   100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
   advertising:    100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
```

route: configuração de rotas estáticas

O comando **route** define rotas estáticas, entradas de tabelas explícitas que jamais mudam (assim esperamos), mesmo se executarmos um daemon de roteamento.[22] Ao adicionarmos uma nova máquina a uma rede local, normalmente precisamos especificar somente a rota-padrão; veja a próxima seção para obter os detalhes.

A discussão sobre roteamento neste livro é dividida entre esta seção e o Capítulo 14, *Roteamento*. Embora grande parte das informações básicas sobre roteamento e o comando **route** esteja aqui, talvez você ache interessante ler as primeiras seções do Capítulo 14 caso precise de mais informações.

O roteamento é realizado na camada IP. Quando chega um pacote vinculado a algum outro host, o endereço IP de destino do pacote é comparado com as rotas na tabela de roteamento do kernel. Se ela coincidir total ou parcialmente com uma rota da tabela, o pacote é encaminhado para o endereço IP do "gateway de próximo salto" associado a esta rota.

Há dois casos especiais. Primeiramente, um pacote pode ser destinado a algum host numa rede diretamente conectada. Nesse caso, o endereço de "gateway de próximo salto" na tabela de roteamento será uma das próprias interfaces do host local, e o pacote é enviado diretamente a seu destino. Esse tipo de rota é adicionado à tabela de roteamento pelo comando **ifconfig** ao configurarmos uma interface.

Em segundo lugar, pode ser que nenhuma rota coincida com o endereço de destino. Nesse caso, a rota-padrão é chamada caso exista alguma. Caso contrário, será retornada uma mensagem ICMP "network unreachable" ou "host unreachable" (rede ou host inatingíveis) ao emitente. Muitas redes locais possuem somente um caminho, e sua rota padrão aponta para ele. No backbone Internet, os roteadores não possuem rotas-padrão — o negócio pára por aí mesmo. Se eles não possuem uma entrada de roteamento para um destino, esse destino não pode ser atingido.

Cada comando **route** acrescenta ou elimina uma rota. O formato é

```
route [op] [type] destination gw gateway [metric] [dev interface]
```

O argumento *op* deve ser **add** para adicionar-se uma rota, **del** para eliminar uma rota, e deve ser omitido para exibir as tabelas de roteamento. *destination* pode ser um endereço de host (digite **-host**), um endereço de rede (digite **-net**) ou então a palavra-chave **default**. Se *destination* for um endereço de rede, você também deve especificar uma máscara de rede.

gateway é a máquina para a qual os pacotes devem ser encaminhados. Ela *deve obrigatoriamente* estar numa rede diretamente conectada; o encaminhamento pode ser realizado somente um salto por vez. O Linux permite que

22. Entretanto, algumas versões do daemon **routed** sobregravarão as rotas estáticas.

Snow666

especifiquemos uma interface em vez do (ou juntamente com o) *gateway*. A palavra-chave **dev** na especificação interface é opcional e pode ser omitida.

metric é o número de encaminhamentos (a contagem de saltos) necessárias para se atingir o destino. O Linux não requer nem usa a contagem de saltos, porém mantém o valor nas tabelas de roteamento se você a ativar, para que os protocolos de roteamento possam usá-las.

O argumento opcional *type* suporta rotas de host, que se aplicam a um endereço IP completo (um host específico) em vez de um endereço de rede. Os valores **-net** e **-host** são aceitos para o parâmetro *type*. Se não for especificado um tipo (*type*), **route** verificará a porção host do endereço de destino para ver se ela é zero. Se a porção host for 0 ou o endereço for uma rede definida no arquivo **/etc/networks**, então a rota é suposta como de uma rede normal.[23]

Já que **route** não pode saber num passe de mágica que números de rede foram divididos em sub-redes, freqüentemente temos de usar o campo *type* para instalar certas rotas. Por exemplo, o endereço 128.138.243.0 se refere a uma rede classe B dividida em sub-rede em nossa instalação, porém para o comando **route** ela se parece como um endereço classe B de 128.138 com uma porção host 243.0; temos de especificar a opção **-net** para eliminar essa confusão feita por **route**. Geralmente, é uma boa prática fornecer um tipo (*type*) explícito para todas as rotas que envolvam sub-redes.

route del *destination* elimina uma entrada específica da tabela de roteamento. Outros sistemas UNIX possuem uma opção para **route**, usualmente **-f** ou **-flush**, que descarregam completamente as tabelas de roteamento e reiniciam. O Linux não suporta essa opção, de forma que talvez você se depare com muitos **route del**s para limpar uma grande tabela de roteamento — certifique-se de estar logado localmente ou talvez você acabe não terminando o serviço e desconectado!

Para inspecionar rotas existentes, use o comando **netstat -nr** ou **netstat -r** caso queira ver os nomes em vez de números. Os números normalmente são melhores caso estejamos depurando, já que a pesquisa de nome pode ser o item problemático.

```
redhat% netstat -nr
Kernel IP routing table
Destination   Gateway         Genmask         Flags   MSS   Window  irtt  Iface
192.168.1.0   0.0.0.0         255.255.255.0   U         0   0          0  eth0
127.0.0.0     0.0.0.0         255.0.0.0       U         0   0          0  lo
0.0.0.0       192.168.1.254   0.0.0.0         UG        0   0          0  eth0

redhat% netstat -r
Kernel IP routing table
Destination   Gateway         Genmask         Flags   MSS   Window  irtt  Iface
192.168.1.0   *               255.255.255.0   U         0   0          0  eth0
127.0.0.0     *               255.0.0.0       U         0   0          0  lo
default       sprint-gw       0.0.0.0         UG        0   0          0  eth0
```

Genmask é a máscara de rede associada ao destino. Flags especificam o estado da rota, como ela foi descoberta e outros parâmetros. Finalmente, Iface é a interface pela qual os pacotes usando essa rota são enviados. Esses exemplos são de um sistema Red Hat, porém o SuSE e o Debian são idênticos exceto pelo fato de o Debian não exibir a rota loopback como padrão.

Rotas-padrão

Uma rota-padrão faz com que todos os pacotes cuja rede de destino não é encontrada na tabela de roteamento do kernel sejam enviados para o gateway indicado. Para configurar uma rota-padrão, basta adicionar a seguinte linha a seus arquivos de inicialização:

```
route add default gw gateway-IP-address
```

Em vez de colocar um endereço IP explícito definido diretamente nos arquivos de inicialização, a maioria dos fornecedores faz com que seus sistemas obtenham o endereço IP de gateway de um arquivo de configuração. A maneira pela qual informações de roteamento locais são integradas na seqüência de inicialização infelizmente é diferente para cada um de nossos sistemas Linux (vamos logo LSB[24], corrija esta síndrome não inventada aqui!).

---

23. **/etc/networks** pode ser usado para associar nomes a números de rede de maneira muito similar àquela como o arquivo **/etc/hosts** associa nomes de hosts a endereços IP completos. Muitos comandos que esperam um número de rede podem aceitar um nome caso ele esteja listado no arquivo **/etc/networks** (ou em DNS).
24. LSB é o Linux Standard Base, uma iniciativa de padronização para o Linux; veja o Capítulo 29 para obter maiores detalhes.

Snow666

A Tabela 13.11 sintetiza as mágicas necessárias.

Tabela 13.11 Como configurar a rota-padrão.

Sistema	Arquivo a ser alterado	Variável a ser alterada
Red Hat	**/etc/sysconfig/network**	`GATEWAY`
SuSE	**/etc/route.conf**	add line: `default` *IP-addr mask interface*
Debian	**/etc/network/interfaces**	gateway

Configurando DNS

Para configurar uma máquina como um cliente DNS, basta editar um ou dois arquivos: todos os sistemas requerem que **/etc/resolv.conf** seja modificado e outros requerem que também modifiquemos um arquivo de "comutação de serviços".

O arquivo **/etc/resolv.conf** lista os domínios DNS que devem ser pesquisados para determinar os nomes que estão incompletos (isto é, não totalmente qualificados, como anchor em vez de anchor.cs.colorado.edu) e os endereços IP dos servidores de nomes a contactar para pesquisas de nomes. A seguir veja uma amostra; para maiores detalhes, veja a seção 16.8.

```
search cs.colorado.edu colorado.edu
nameserver 128.138.242.1
nameserver 128.138.243.151
nameserver 192.108.21.1
```

/etc/resolv.conf deve listar primeiramente o servidor de nomes estável "mais próximo", pois o servidor na primeira posição será contactado primeiro. Podemos ter até três entradas do tipo **nameserver**. Se possível, devemos sempre ter mais de uma. O período de timeout para uma consulta DNS a um determinado servidor de nomes parece ser bem longo, de modo que se o primeiro servidor de nomes não responder, seus usuários perceberão.

Algumas vezes você verá a linha `domain` em vez de uma linha `search`. Tal linha indica um arquivo **resolv.conf** antigo que não foi atualizado para usar a diretiva `search` ou um determinador antigo que não entende `search`. `domain` e `search` não são equivalentes; `search` é o preferido. Até esta data, o Red Hat usa como padrão um arquivo **resolv.conf** que utilizar `domain` em vez de `search`.

Veja o Capítulo 18 para obter maiores informações sobre NIS.

Alguns sistemas não usam DNS como padrão, mesmo se existir um arquivo **resolv.conf** configurado apropriadamente. Esses sistemas possuem um arquivo comutador de serviços que determina que mecanismos serão usados para determinar as associações nome de host-endereço IP. A priorização de fontes de informação é vista de forma mais detalhada a partir da seção 18.3, porém mencionamos o tópico aqui também, já que ele algumas vezes frustra suas tentativas de configurar uma máquina nova.

O arquivo comutador de serviços lhe permite especificar a ordem na qual DNS, NIS e **/etc/hosts** devem ser consultados. Na maioria dos casos, você também pode excluir inteiramente certas fontes de dados. Sua opção de ordem impacta a capacidade da máquina de inicializar e a maneira pela qual esta inicialização interage com o conteúdo do arquivo **/etc/hosts**.

Se DNS for escolhido como a primeira fonte de dados a ser consultada, talvez seja necessário ter um servidor de nomes na rede local e ter o seu nome de host e o endereço IP no arquivo **hosts** para tudo funcionar no momento da inicialização.

A Tabela 13.12 apresenta a localização dos arquivos de configuração relevantes e a configuração-padrão para pesquisas de host em cada um de nossos sistemas escolhidos como exemplo.

Tabela 13.12 Arquivos comutadores de serviços por sistema.

Sistema	Arquivo comutador[a]	Padrão para pesquisas de nomes de hosts
Red Hat	**/etc/nsswitch.conf** **/etc/host.conf**	`files nisplus nis dns` `hosts, bind`
SuSE e Debian	**/etc/nsswitch.conf** **/etc/host.conf**	`files dns` `hosts, bind`

a. A maioria dos aplicativos são associados a **libc6**, que usa o determinador do BIND e **nsswitch.conf**. Alguns aplicativos mais antigos são associados a **libc4** ou **libc5**, que usa **host.conf**.

www.snow666.blogspot.com
Snow666

Veja o Capítulo 22 para obter maiores informações sobre interfaces virtuais.

A pilha de rede do Linux

A pilha de rede nos kernels 2.2 e posteriores do Linux suporta interfaces de rede virtuais, SACKs (Selective ACKnowledgements, confirmações seletivas), bem como um novo recurso IP, ECN (Explicit Congestion Notification, notificação de congestionamento explícita). O ECN marca pacotes TCP para notificar o emitente sobre congestionamento em vez de deixar a recuperação de pacotes perdidos, que jamais chegaram ao recebedor, servir como a única indicação de problemas. O ECN foi especificado originalmente na RFC2481 (janeiro de 1999) e agora é um padrão proposto, documentado na RFC3168. A RFC2884 (julho de 2000) inclui dados de desempenho e uma avaliação de ECN. Em geral, foi determinado que o ECN é uma coisa boa tanto para transferências massivas de dados quanto para dados transacionais como solicitações e respostas Web.

O Linux sempre é uma das primeiras pilhas de rede a incluir recursos novos. Algumas vezes, o pessoal do Linux é tão rápido que o restante da infra-estrutura de rede não é capaz de interoperar. Por exemplo, o recurso ECN do Linux (que é ativo como padrão) colidiu com ajustes de configuração padrão incorretos num produto firewall da Cisco, fazendo com que todos os pacotes com o bit ECN ativo fossem perdidos. Oops!

A implementação Linux original de confirmações seletivas estava errada ou quem sabe, mais precisamente, não otimizada. Quando, por exemplo, um SACK chega e especifica que os pacotes 2, 5 e 6 foram perdidos, o comportamento apropriado do emitente é reenviar todos os três pacotes imediatamente para facilitar uma rápida recuperação. O Linux envia os três pacotes apropriados, porém somente um de cada vez, aguardando que cada pacote seja confirmado antes de prosseguir. É raro para o Linux ser acusado de muito conservador, mas no caso de confirmações seletivas, isto é verdade.

13.8 Configuração de Redes Específicas de Cada Distribuição

O Capítulo 2 descreve os detalhes suculentos dos procedimentos de inicialização das distribuições apresentadas como exemplo neste livro. Nas próximas seções, simplesmente sintetizamos as tarefas que estão relacionadas à configuração de uma rede. Os sistemas aqui apresentados configuram a interface loopback automaticamente; você jamais terá de modificar esta parte da configuração. Além disso, cada sistema difere um pouco um do outro.

Uma alternativa para a configuração manual é **linuxconf**, um utilitário baseado em módulos que fornece uma interface simples para gerenciamento de uma série de tarefas para administração de sistemas, inclusive a maior parte da configuração relacionada com rede. Há três interfaces com o usuário: baseada em texto, Web e X Windows. Digitamos parâmetros locais para diversas tarefas de configuração, e **linuxconf** faz a configuração para nós.

linuxconf pode ser encontrado gratuitamente na Web, mas não é incluso na instalação-padrão. www.solucorp.qc.ca/linuxconf possui um demonstrativo baseado na Web que é executado num ambiente onde foi aplicado **chroot** de modo que você poderá experimentá-lo sem danificar os seus sistemas.

Quatro arquivos são comuns a cada um dos nossos sistemas abordados: **/etc/hosts**, **/etc/resolv.conf**, **/etc/nsswitch.conf** e **/etc/host.conf**. Estes foram abordados nas seções de configuração de rede genérica anteriores e, exceto por **resolv.conf** e possivelmente **hosts**, usualmente não precisam ser modificados quando acrescentamos uma máquina à rede.

Após qualquer modificação num arquivo que controla a configuração de rede no momento do boot, talvez você também precise reinicializar ou então desativar a interface de rede e a seguir ativá-la novamente para que a alteração entre em vigor. No Red Hat ou Debian você pode usar os comandos **ifup** e **ifdown**; no SuSE, basta reinicializar a máquina.

Configuração de Rede para o Red Hat

A Tabela 13.13 ilustra os arquivos de configuração do Red Hat para parâmetros de rede.

Tabela 13.13 Arquivos de configuração de rede no Red Hat.

Arquivo	O que é configurado no arquivo
/etc/sysconfig/network	Hostname, rota-padrão
/etc/sysconfig/static-routes	Rotas estáticas
network-scripts/ifcfg- *ifname*[a]	Endereço IP, máscara de rede, endereço de difusão ampla por interface

a. Relativo a **/etc/sysconfig**.

Snow666

O nome de host de uma máquina Red Hat é configurado no arquivo **/etc/sysconfig/network**, que também contém linhas que especificam o gateway-padrão e domínio DNS da máquina. Por exemplo, eis um arquivo **network** para um host que possui uma única interface Ethernet:

```
NETWORKING=yes
HOSTNAME=redhat.toadranch.com
DOMAINNAME=toadranch.com        ### optional
GATEWAY=192.168.1.254
```

Dados específicos a uma interface que são armazenados em **/etc/sysconfig/network-scripts/ifcfg-*ifname***, onde *ifname* é o nome da interface de rede. Esses arquivos de configuração nos permitem configurar o endereço IP, máscara de rede, endereços de rede e de difusão ampla para cada interface. Eles também incluem uma linha que especifica se a interface deve ser configurada como "ativa" no momento do boot.

Tipicamente, arquivos para uma interface Ethernet (eth0) e para a interface loopback (lo) estão presentes. Por exemplo,

```
DEVICE=eth0
IPADDR=192.168.1.13
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
ONBOOT=yes
```

e

```
DEVICE=lo
IPADDR=127.0.0.1
NETMASK=255.0.0.0
NETWORK=127.0.0.0
BROADCAST=127.255.255.255
ONBOOT=yes
NAME=loopback
```

são os arquivos **ifcfg-eth0** e **ifcfg-lo** para a máquina redhat.toadranch.com descrita no arquivo **network** anteriormente nesta seção.

O Red Hat fornece uma série de scripts práticos que facilitam o gerenciamento de interfaces. **ifup** e **ifdown** aceitam o nome de uma interface de rede como argumento e ativam ou desativam a interface especificada. Após alterar as informações de rede em qualquer um dos diretórios **/etc/sysconfig**, certifique-se de executar **ifdown** *ifname* seguido de **ifup** *ifname*. Melhor ainda, reinicialize o sistema para certificar-se de que suas alterações não provoquem nenhum tipo de problema sutil. Não há nenhuma página de manual para **ifup** e **ifdown**, porém eles são scripts de shell (mantidos em **/sbin**), de modo que você pode dar uma olhada e verificar o que eles fazem detalhadamente.

Se você precisar gerenciar todas as interfaces de uma só vez, o script **/etc/rc.d/init.d/network** aceita os argumentos **start**, **stop**, **restart** e **status**. Esse script é chamado no momento do boot com o argumento **start**.

Os scripts do Red Hat também são capazes de configurar rotas estáticas. Qualquer rota adicionada ao arquivo **/etc/sysconfig/static-routes** é introduzida na tabela de roteamento no momento do boot. As entradas fornecem argumentos a **route add**, embora numa ordem misturada (a interface está em primeiro lugar em vez de último):

```
eth0 net 130.225.204.48 netmask 255.255.255.248 gw 130.225.204.49
eth1 net 192.38.8.0 netmask 255.255.255.224 gw 192.38.8.129
```

A interface é especificada primeiro, seguida por argumentos para **route**: o tipo de rota (`net` ou `host`), a rede de destino, a máscara de rede associada a esta rede e, finalmente, o gateway do próximo salto. A palavra-chave `gw` é exigida. Os kernels Linux atuais não usam o parâmetro métrico (`metric`) para **route**, porém permitem que ele seja introduzido e mantido na tabela de roteamento para que os daemons de roteamento o usem. O exemplo **static-routes** acima produziria os seguintes comandos **route**:

```
route add -net 130.225.204.48 netmask 255.255.255.248 gw 130.225.204.49 eth0
route add -net 192.38.8.0 netmask 255.255.255.224 gw 192.38.8.129 eth1
```

Configuração de rede para o SuSE

A Tabela 13.14 ilustra os arquivos de configuração de rede para o Linux SuSE.

Snow666

Tabela 13.14 Arquivos de configuração de rede SuSE.

Arquivo	O que é configurado no arquivo
/etc/rc.config	Nome de host, endereço IP, máscara de rede e outros
/etc/route.conf	Rota-padrão, rotas estáticas

O Linux SuSE configura a maioria das opções de configuração, inclusive aquelas relacionadas com rede, no arquivo **/etc/rc.config**. Ele também fornece uma ferramenta de configuração, **/sbin/SuSEconfig**, que usa scripts em **/sbin/conf.d** e **/etc/rc.config.d** para fazerem na retaguarda todos os tipos de coisas relacionadas com configurações sensíveis e que merecem cuidado especial.

/etc/rc.config contém todos os parâmetros relacionados com redes, exceto informações de roteamento e informações DNS. No trecho abaixo, acrescentamos um comentário introduzido por ### para explicar as linhas menos óbvias. O próprio arquivo **rc.config** é formado em grande parte por linhas de comentário e realiza um excelente trabalho de autodocumentação.

```
START_LOOPBACK="yes"      ### start loopback pseudo-device?
NETCONFIG="_0"            ### #of net cards; _0 = 1, _1 = 2, etc.
IPADDR_0="192.168.1.101"
NETDEV_0="eth0"           ### logical name of first network device
IFCONFIG_0="192.168.1.101 broadcast 192.168.1.255 netmask 255.255.255.0"
FQHOSTNAME="inura.toadranch.com"
DISABLE_ECN="yes"         ### TCP Explicit Congestion Notification
```

Interfaces Ethernet múltiplas teriam seus parâmetros configurados em variáveis com os sufixos _1, _2 e _3; por exemplo, IPADDR_1, NETDEV_1, IFCONFIG_1. Como padrão são suportadas até quatro interfaces.

Informações de roteamento estático para um sistema SuSE são armazenadas no arquivo **/etc/route.conf**. Por exemplo, para o host configurado acima, que possui somente uma rota-padrão, ele contém

```
default 192.168.1.254 0.0.0.0 eth0
```

onde 192.168.1.254 é o endereço IP do host gateway padrão na rede local atingida através de **/dev/eth0**. O formato de **route.conf** é o mesmo que os argumentos para o comando **route**: destino, gateway, máscara de rede, interface e parâmetros extras opcionais armazenados na tabela de roteamento para emprego de daemons de roteamento.

Roteamento dinâmico é configurado pelo arquivo **rc.config**, usando as variáveis mostradas a seguir (com seus valores-padrão). Adicionamos comentários para identificá-las.

```
START_ROUTED="no"  ### daemon RIP (Routing Information Protocol, protocolo de informações de
                   ### roteamento) versão 1
START_ZEBRA="no"   ### gerenciador de roteamento zebra
START_BGPD="no"    ### daemon BGP (Border Gateway Protocol, protocolo de gateway de fronteira)
START_RIPNG="no"   ### daemon RIP versão 2
START_OSPFD="no"   ### daemon de protocolo OSPF (Open Shortest Path First)
START_MRTD="no"    ### daemon de roteamento multitarefa baseado em threads
```

Configuração de rede para o Debian

Conforme ilustrado na Tabela 13.15, a configuração de rede para o Debian ocorre em grande parte nos arquivos **/etc/hostname** e **/etc/network/interfaces**, com um pouco de ajuda do arquivo **options** no diretório **/etc/network**.

Table 13.15 Arquivos de configuração de rede no Debian.

Arquivo	O que é configurado no arquivo
/etc/hostname	Nome do host
/etc/network/interfaces	Endereço IP, máscara de rede, rota-padrão
/etc/network/options	Opções de rede de baixo nível (encaminhamento IP etc.)

O nome do host é configurado em **/etc/hostname**. O nome nesse arquivo deve ser totalmente qualificado, pois o seu valor é usado em outros contextos em que se espera um nome totalmente qualificado. Porém, a instalação De-

Snow666

bian padrão deixa um nome não qualificado aí. O endereço IP, a máscara de rede e o gateway-padrão são configurados no arquivo **/etc/network/interfaces**.

Por exemplo:

```
iface lo inet loopback
iface eth0 inet static
      address 192.168.1.102
      netmask 255.255.255.0
      gateway 192.168.1.254
```

Os comandos **ifup** e **ifdown** lêem esse arquivo e acionam ou desativam as interfaces chamando comandos de nível mais baixo como **ifconfig** com os parâmetros corretos. O formato básico do arquivo **interfaces** inclui uma linha começando com a palavra-chave `iface` para cada interface seguida por linhas recuadas que especificam mais parâmetros.

A palavra-chave `inet` na linha `iface` é a família de endereço; esta sempre será `inet`. A palavra-chave `static` é chamada de um método e especifica que o endereço IP e a máscara de rede para eth0 serão atribuídos diretamente. As linhas `address` e `netmask` são necessárias para configurações estáticas; versões anteriores do kernel Linux também exigiam que fosse especificado o endereço de rede, porém agora o kernel é mais inteligente e capaz de descobrir o endereço de rede pelo endereço IP e pela máscara de rede. A linha `gateway` especifica o endereço do gateway-padrão e é usada para instalar uma rota-padrão.

O arquivo **options** permite que algumas variáveis de rede sejam configuradas no momento da inicialização. Como padrão, o Debian deixa desativado o encaminhamento IP, ativada a proteção contra spoofing e desativada a proteção contra syn cookies.

Configuração de rede com uma Interface Gráfica com o Usuário

Descrevemos detalhadamente como configurar interfaces de rede na linha de comandos. Caso você provenha do mundo Windows e prefira apontar e clicar em vez de digitar, considere as diversas interfaces gráficas com o usuário que fornecedores disponibilizam para tratar de configuração de rede. Muitas parecem suspeitas como a configuração de rede Windows, porém sem a etapa "Você tem de reinicializar o seu computador ...".

Por exemplo, o Red Hat inclui uma ferramenta chamada **neat** (NEtwork Administration Tool) capaz de realizar configurações Ethernet, modem, ISDN, xDSL e wireless. Para executá-la, selecione

```
Main menu --> programs --> system --> network configuration
```

da área de trabalho GNOME ou simplesmente digite **neat** num shell. **neat** substitui a tentativa anterior do Red Hat de uma ferramenta de configuração com interface gráfica com o usuário, **netcfg**.

13.9 DHCP

O DHCP é definido nas RFCs 2131 e 2132.

Os hosts Linux sempre exigiram configuração manual para serem adicionados a uma rede. Ao conectar um Mac ou PC numa rede, ele simplesmente funciona. Por que o Linux não pode fazer o mesmo? O DHCP (Dynamic Host Configuration Protocol, protocolo de configuração de host dinâmico) torna essa expectativa razoável muitos passos mais próximo da realidade.

O protocolo habilita um cliente DHCP para arrendar uma variedade de parâmetros de rede e administrativos de um servidor central que está autorizado a distribuí-los. O paradigma de arrendamento é particularmente conveniente para PCs que são desligados quando não estão em uso e para ISPs que possuem clientes discados intermitentes.

Entre os parâmetros arrendáveis temos:

- Endereços IP e máscaras de rede
- Gateways (rotas-padrão)
- Servidores de nomes DNS
- Hosts para syslog
- Servidores WINS, servidores de fontes X, servidores proxy, servidores NTP
- Servidores TFTP (para carregar uma imagem de boot)

e muitos outros mais (veja a RFC2132). Entretanto, o emprego no mundo real de parâmetros mais exóticos é raro. Em muitos casos, um servidor DHCP fornece somente parâmetros de rede básicos como endereços IP, máscaras de rede, gateways-padrão e servidores de arquivos de nomes.

Snow666

Clientes têm de se reportar periodicamente ao servidor DHCP para renovar os seus arrendamentos. Se um arredamento não for renovado, ele eventualmente expira. O servidor DHCP então fica livre para atribuir o endereço (ou seja lá o que estiver sendo arrendado) a um cliente diferente. O período de arrendamento é configurável, porém normalmente é bem longo (horas ou dias).

O DHCP é capaz de poupar um bom tempo e sofrimento de um administrador de sistemas outrora desafortunado. Assim que o servidor estiver ativo e em funcionamento, os clientes poderão usá-lo para obter automaticamente suas configurações no momento do boot. Sem frescuras, sem confusão.

Software DHCP

A Tabela 13.16 ilustra o software DHCP que é expedido com sistemas que adotamos como exemplos neste livro.

Tabela 13.16 Software DHCP nos sistemas adotados neste livro.

Sistema	Cliente DHCP	Servidor DHCP
Red Hat	**/usr/sbin/dhcpcd** e **/sbin/pump**	Disponível na forma de um pacote RPM
Suse	**/sbin/dhcpcd**	**/usr/sbin/dhcpd** de ISC
Debian	**/sbin/pump** com a versão 2.2	Disponível com **apt-get**
	/sbin/dhclient com a versão 3.0	

O ISC (Internet Software Consortium) criou uma implementação de referência do protocolo DHCP. O servidor, o cliente e o agente de repasse se encontram disponíveis em ftp.isc.org. O servidor ISC também fala o protocolo BOOTP, que é similar conceitualmente ao DHCP, porém menos sofisticado.

Recomendamos o pacote ISC em relação a todas as implementações específicas dos fornecedores. Num ambiente de rede heterogêneo típico, a administração pode ser simplificada em muito por padronização em uma única implementação. O software ISC oferece uma solução confiável e de fonte aberto que é acoplada sem incidentes na maioria dos sistemas. A versão 3.0 contém muitas opções de configuração novas e úteis, inclusive comportamento condicional, pools de endereços distintos com controle de acesso, atualizações DNS dinâmicas e outros.

Os clientes DHCP têm de iniciar uma conversação com o servidor DHCP usando o endereço de difusão ampla genérico todo preenchido somente com 1s — os clientes não conhecem ainda suas máscaras de sub-rede e, conseqüentemente, não podem usar o endereço de difusão ampla de sub-rede.

O servidor DHCP do ISC fala o protocolo de atualização dinâmica DNS. O servidor não somente dá o seu host o seu endereço IP e outros parâmetros de rede, como também pode atualizar o banco de dados DNS com o mapeamento nome de host-endereço IP correto. Veja o Capítulo 16 para obter maiores informações sobre atualizações DNS dinâmicas.

Nas seções a seguir, discutiremos brevemente o protocolo DHCP, explicaremos como configurar o servidor ISC que o implementa e, a seguir, discutiremos algumas questões de configuração de clientes.

Funcionamento do DHCP

O DHCP é uma extensão do BOOTP compatível com versões anteriores, um protocolo que era originalmente destinado a permitir que estações de trabalho UNIX sem disco fossem capazes de fazer o boot. BOOTP fornece aos clientes seus endereço IP, máscara de rede, gateway-padrão e informação de inicialização TFTP. O DHCP generaliza os parâmetros que podem ser fornecidos e acrescenta o conceito de "arrendamento".

Um cliente DHCP inicia sua interação com um servidor DHCP transmitindo a todos uma mensagem de "Socorro! Quem sou eu?". Se houver um servidor DHCP na rede local, ele negocia com o cliente para arrendar-lhe um endereço IP e fornecer-lhe outros parâmetros de rede (máscara de rede, informações de servidor de nomes e gateway-padrão). Se não houver nenhum servidor DHCP na rede local, os servidores em sub-redes diferentes também poderão receber a mensagem inicial de um proxy chamado de "agente relay".

Quando o tempo de arrendamento do cliente estiver pela metade, ele renovará o arrendamento. O servidor é obrigado a controlar os endereços que ele manipulou e essas informações têm de persistir entre reinicializações. Supostamente os clientes também têm de manter seus estados de arrendamento entre reinicializações, embora muitos não o façam. O objetivo é maximizar a estabilidade na configuração da rede.

Por sinal, o DHCP normalmente não é usado para configurar dial-up PPP interfaces. O próprio PPPCP (PPP Control Protocol, protocolo de controle do PPP) do PPP normalmente desempenha esse papel.

Snow666

Servidor DHCP do ISC

O servidor DHCP do ISC pode ser encontrado em ftp.isc.org ou então em www.isc.org. Os detalhes a seguir referem-se à versão 2 do pacote (incluída na distribuição SuSE). A versão 3.0 está por vir, de forma que você deve se certificar sobre qual versão efetivamente fez o download para usar essas instruções.

Descompacte o arquivo **tar.gz** e dê o comando **cd** para ir para o diretório da distribuição. Você deve ver subdiretórios para o servidor, o cliente e o agente relay, juntamente com um diretório de código compartilhado. Execute **./configure** seguido de **make** e **make install** para construir e instalar cada um desses módulos.

O DHCP do Debian é o ISC DHCP, de modo que você pode simplesmente aplicar o comando **apt-get install dhcp dhcp-relay** para obter o cliente, o servidor e o agente relay.

Para configurar o servidor DHCP, **dhcpd**, você precisa editar o arquivo **dhcpd.conf** modelo do diretório **server** e instalá-lo em **/etc/dhcpd.conf**.[25] Você também tem de criar um arquivo vazio de banco de dados de arrendamento chamado **/var/db/dhcp.leases**; use o comando **touch**. Certifique-se de que o **dhcpd** seja capaz de efetuar gravações nesse arquivo. Para configurar o arquivo **dhcpd.conf**, você precisará das seguintes informações:

- As sub-redes para as quais o **dhcpd** deve dirigir endereços IP e os intervalos de endereços a serem doados
- As durações inicial e máxima do arrendamento, medidas em segundos
- Configurações para clientes BOOTP caso você tenha alguma (eles possuem endereços IP estáticos e também têm de ter listados seus endereços de hardware em nível MAC)
- Quaisquer outras opções que o servidor tenha de passar a clientes DHCP: máscara de rede, rota-padrão, domínio DNS, servidores de nomes etc.

A página de manual de **dhcpd** lhe dá uma visão geral do processo de configuração. A sintaxe exata do arquivo de configuração é vista na página de manual **dhcpd.conf**. Ambas se encontram no subdiretório **server** da distribuição. O Linux SuSE inclui um modelo de arquivo **dhcpd.conf** no diretório **/etc**; modifique-o para ficar de acordo com a configuração de rede de sua instalação local.

dhcpd deve ser iniciado automaticamente no momento do boot. Talvez você ache interessante fazer a inicialização do daemon condicionada à existência de **/etc/dhcpd.conf**.

Eis um exemplo de arquivo **dhcpd.conf** de um computador Linux com duas interfaces: uma interna e uma que se conecta à Internet. Essa máquina realiza conversões NAT para a rede interna e também arrenda um intervalo de 10 endereços IP nesta rede. O arquivo **dhcpd.conf** contém uma entrada fictícia para a interface externa (obrigatória) e uma entrada host para uma determinada máquina que precisa de um endereço fixo.

```
# dhcpd.conf
#
# global options
option domain-name "synack.net";
option domain-name-servers gw.synack.net;
option subnet-mask 255.255.255.0;
default-lease-time 600;
max-lease-time 7200;

subnet 192.168.1.0 netmask 255.255.255.0 {
   range 192.168.1.51 192.168.1.60;
   option broadcast-address 192.168.1.255;
   option routers gw.synack.net;
}

subnet 209.180.251.0 netmask 255.255.255.0 {
}
host gandalf {
   hardware ethernet 08:00:07:12:34:56;
   fixed-address gandalf.synack.net;
}
```

25. Tome cuidado: o formato do arquivo **dhcpd.conf** é um tanto frágil. O simples fato de omitir um ponto-e-vírgula já é o bastante para você receber uma mensagem de erro vaga e inútil.

Snow666

Veja o Capítulo 16 para obter maiores informações sobre o DNS.

Endereços atribuídos pelo DHCP podem eventualmente entrar em conflito com o conteúdo do banco de dados DNS. Os sites normalmente atribuem um nome genérico a cada um dos endereços arrendados dinamicamente (por exemplo, dhcp1.synack.net) e permitem que os nomes de máquinas particulares "flutuem" juntamente com os seus endereços IP. Se estiver usando uma versão recente do BIND que ofereça suporte a atualizações dinâmicas, também poderá configurar **dhcpd** para atualizar o banco de dados DNS à medida que ele manipula endereços. A solução de atualização dinâmica é mais complicada, porém possui a vantagem de preservar o nome de host de cada máquina.

dhcpd registra cada transação de arrendamento no arquivo **dhcp.leases**. Ele também faz um backup periódico deste arquivo renomeando-o para **dhcpd.leases~** e recriando o arquivo **dhcp.leases** de seu banco de dados residente em memória. Se **dhcpd** travasse durante essa operação, você poderia acabar ficando apenas com um arquivo **dhcp.leases~** file. Nesse caso, **dhcpd** se recusará a iniciar e você terá de renomear o arquivo antes de reiniciá-lo. *Não* crie simplesmente um arquivo **dhcp.leases** vazio. Caso contrário, o caos se instalará, já que os clientes acabarão ficando com endereços duplicados.

Configuração DHCP no Red Hat

O Red Hat costumava vir com o servidor ISC DHCP, **dhcpd**, mas isso não é mais verdade na situação-padrão. Ele ainda se encontra disponível como pacote e seu download pode ser feito em um dos sites-espelho do Red Hat. O Red Hat efetivamente vem com dois clientes DHCP diferentes: **pump**[26] e **dhcpcd**. Caso esteja ficando enfastiado, sugerimos que você ignore ambos os clientes DHCP fornecidos e instale o cliente do Internet Software Consortium em www.isc.org. Por experiência própria, ele é mais confiável, porém algumas vezes um desafio em termos de compilação (veja as dicas na distribuição Linux).

pump é o cliente DHCP padrão do Red Hat. Ele é inicializado no momento do boot quando solicitado por um dos fragmentos de script **/etc/sysconfig/network-scripts/ifcfg-*interface***. Por exemplo, para configurar a interface eth0 automaticamente com o DHCP, você teria de editar o arquivo **/etc/sysconfig/network-scripts/ifcfg-eth0**. Em vez de configurar o endereço IP da interface, a máscara de rede e outros parâmetros, você incluiria a linha

```
BOOTPROTO=dhcp
```

Caso precise iniciar **pump** manualmente para gerenciar eth0, execute o comando a seguir:

```
# pump -i eth0
```

Para interromper o gerenciamento de eth0 por parte de **pump**, use

```
# pump -r -i eth0
```

dhcpcd é a alternativa fornecida para **pump**, mas para usá-lo automaticamente você tem de substituir chamadas a **pump** por chamadas a **dhcpcd** nos scripts de inicialização **ifup** e **ifdown**. **dhcpcd** é configurado por arquivos no diretório **/etc/dhcpc**, que é vazio na versão atual do Red Hat. Também há um diretório **/etc/dhcpcd** que contém um arquivo que é linkado simbolicamente a si mesmo. Não estamos certos do que realmente se pretendia nesse caso, porém, de qualquer maneira, isso parece não ser de muita utilidade. A arte do desempenho, quem sabe?

Configuração DHCP no SuSE

O SuSE vem com o servidor DHCP do ISC, **/usr/sbin/dhcpd** e um cliente DHCP, **/sbin/dhcpcd**, escritos por Yoichi Hariguchi e Sergei Viznyuk.

A instalação e a configuração do servidor ISC foram vistas brevemente aqui. Configure as seguintes variáveis de **/etc/rc.config** para iniciar automaticamente o servidor DHCP. Use o agente de repasse caso precise cobrir várias redes com o mesmo servidor.

```
START_DHCPD="no"            ### servidor DHCP, não inicializado, na situação-padrão
DHCPD_INTERFACE="eth0"      ### interface do servidor
START_DHCRELAY="no"         ### agente relay DHCP não inicializado, na situação-padrão
DHCRELAY_SERVERS="127.0.0.1 127.0.0.2" ### endereços IP do agente relay
```

O cliente DHCP realmente não requer configuração. Ele armazena arquivos para conexão no diretório **/var/state/dhcp**. Os arquivos recebem nomes conforme as interfaces que eles descrevem. Por exemplo, **dhcpcd-eth0.info** conteria todos os parâmetros de rede que **dhcpcd** tinha configurado para a interface eth0.

26. Uma sapatilha (pump), assim como uma bota (boot) (como em BOOTP), é um tipo de calçado.

Snow666

A opção **-D** de **dhcpcd** o força a aceitar o nome de domínio do servidor (o padrão é não aceitá-lo). Alguns ISPs exigem o uso da opção **-h** *hostname* para fins de autenticação. Há também uma opção de depuração **-d** que é útil caso você esteja tendo problemas em fazer com que **dhcpcd** funcione corretamente. **dhcpcd** loga através de syslog para a instalação LOCAL0 marcada com o nome do programa **dhcpcd**.

Para solicitar a uma interface que use o protocolo DHCP para obter seus parâmetros de rede, configure a linha IFCONFIG correspondente em **/etc/rc.config** para conter a string "dhcpclient". Por exemplo,

```
IFCONFIG_0="dhcpclient"
```

faria com que a interface eth0 fosse configurada por **dhcpcd**. Também há uma série de outras variáveis relacionadas a DHCP em **rc.config**:

```
DHCLIENT_SLEEP="3"          ### para dar tempo a interfaces "cansadas" de serem  inicializadas
DHCLIENT_SET_HOSTNAME="no"  ### o nome de host deve ser configurado pelo DHCP?
```

Configuração DHCP no Debian

O DHCP é configurado em sistemas Debian como um método (similar à `static`; veja a página 210) no arquivo **/etc/network/interfaces**. O método se chama `dhcp`; ele suporta as opções `hostname`, `leasehours`/`leasetime`, `vendor` e `client`. O Debian inclui tanto o cliente DHCP **pump** quanto o cliente **dhcpcd**. **pump** usa as opções `hostname` e `leasehours`; **dhcpcd** precisa de `hostname`, `leasetime`, `vendor` e `client`. As unidade de `leasetime` são segundos, não horas. Nenhuma destas opções é obrigatória.

No Debian 3.0, **dhcpcd** não existe mais, **pump** é disponível através de **apt-get**, e o cliente DHCP padrão é o **dhclient** ISC.

13.10 Reconfiguração e Ajuste Dinâmico do Linux

O Linux possui sua própria maneira especial de ajustar o kernel e parâmetros de rede; a descreveremos em detalhes para o Red Hat e a seguir apontaremos as diferenças existentes no SuSE e no Debian. Em vez de fornecer um comando que lê e configura os parâmetros, o Linux coloca uma representação de cada variável que pode ser ajustada no sistema de arquivos **/proc**. As variáveis de rede se encontram em **/proc/sys/net/ipv4**:

```
% cd /proc/sys/net/ipv4; ls -F
conf/                           ipfrag_low_thresh      tcp_max_tw_buckets
icmp_destunreach_rate           ipfrag_time            tcp_mem
icmp_echo_ignore_all            ip_local_port_range    tcp_orphan_retries
icmp_echo_ignore_broadcasts     ip_nonlocal_bind       tcp_reordering
icmp_echoreply_rate             ip_no_pmtu_disc        tcp_retrans_collapse
icmp_ignore_bogus_error_responses neigh/               tcp_retries1
icmp_paramprob_rate             route/                 tcp_retries2
icmp_timeexceed_rate            tcp_abort_on_overflow  tcp_rfc1337
igmp_max_memberships            tcp_adv_win_scale      tcp_rmem
inet_peer_gc_maxtime            tcp_app_win            tcp_sack
inet_peer_gc_mintime            tcp_dsack              tcp_stdurg
inet_peer_maxttl                tcp_ecn                tcp_synack_retries
inet_peer_minttl                tcp_fack               tcp_syncookies
inet_peer_threshold             tcp_fin_timeout        tcp_syn_retries
ip_autoconfig                   tcp_keepalive_intvl    tcp_timestamps
ip_default_ttl                  tcp_keepalive_probes   tcp_tw_recycle
ip_dynaddr                      tcp_keepalive_time     tcp_window_scaling
ip_forward                      tcp_max_orphans        tcp_wmem
ipfrag_high_thresh              tcp_max_syn_backlog
```

Muitas dessas variáveis com **rate** e **max** em seus nomes são usadas para impedir ataques de recusas de serviços. O subdiretório **conf** contém variáveis que são configuradas por interface. Ele contém subdiretórios **all** e **default** e um subdiretório para cada interface (inclusive a loopback). Cada subdiretório contém o mesmo conjunto de arquivos.

```
% cd conf/default; ls -F
accept_redirects      bootp_relay     mc_forwarding   secure_redirects tag
accept_source_route   forwarding      proxy_arp       send_redirects
arp_filter            log_martians    rp_filter       shared_media
```

Snow666

Se você alterar alguma coisa no subdiretório **all**, suas alterações se aplicarão a todas as interfaces. Se você modificar a mesma variável no, digamos, subdiretório **eth0**, somente esta interface será afetada. O subdiretório **defaults** contém os valores-padrão conforme sai de fábrica.

O diretório **neigh** também contém um subdiretório para cada interface. Os arquivos em cada subdiretório controlam o gerenciamento de tabelas ARP e a descoberta de vizinhos IPv6 para esta interface em particular. Eis a lista de variáveis; aquelas iniciadas por **gc** (de garbage collection, ou seja, coleta de lixo) determinam como as entradas de tabelas ARP atingem o seu tempo de sobrevida e como são descartadas.

```
% cd neigh/default; ls -F
anycast_delay           gc_interval     gc_thresh3     proxy_qlen
app_solicit             gc_stale_time   locktime       retrans_time
base_reachable_time     gc_thresh1      mcast_solicit  ucast_solicit
delay_first_probe_time  gc_thresh2      proxy_delay    unres_qlen
```

Para observar o valor de uma variável, use **cat**; para configurá-la, use **echo** redirecionado para o nome de arquivo apropriado. Por exemplo, o comando

```
% cat icmp_echo_ignore_broadcasts
0
```

mostra que o valor dessa variável é 0, significando que os pings de difusão ampla não são ignorados. Para configurá-la em 1 (e evitar ser presa de ataques de negação de serviço), execute

```
sudo sh -c "echo 1 > icmp_echo_ignore_broadcasts"[27]
```

no diretório **/proc/sys/net**. Normalmente você estará logado na mesma rede em que está ajustando essas variáveis; portanto, tome cuidado! Você pode bagunçar as coisas de tal maneira a exigir uma reinicialização na console para poder recuperar, o que poderia ser inconveniente se, por acaso, o sistema estivesse em Point Barrow, Alasca, e no pico do inverno. Faça ajustes finos e testes dessas variáveis em seu sistema desktop antes de pensar em mexer numa máquina de produção.

O documento **/usr/src/linux/Documentation/proc.txt**, escrito pelo pessoal do SuSE, é uma boa introdução sobre ajustes de kernels com **/proc**.[28] Ele lhe informa qual o verdadeiro significado das variáveis e, algumas vezes, fornece sugestões para valores. O arquivo **proc.txt** somente pode ser lido por root e, portanto, você terá de usar **sudo** para visualizá-lo[29]. Ele também se encontra um tanto defasado, já que os programadores do código Linux parecem escrever mais rápido do que aqueles que escrevem sua documentação.

O SuSE coloca o arquivo de documentação **proc.txt** mencionado acima no subdiretório **filesystems** de **/usr/src/linux/Documentation**. Uma série de variáveis visíveis no sistema de arquivos **/proc** é configurável em **/etc/rc.config**. Por exemplo:

```
IP_TCP_SYNCOOKIES="yes"
IP_FORWARD="no"
```

O Debian também possui alguns poucos parâmetros configuráveis que são listados nos arquivos de configuração de rede e depois configurados em **/proc** durante a inicialização. Eles são listados no arquivo **/etc/network/options**. Como padrão, ele contém:

```
ip_forward=no
spoofprotect=yes
syncookies=no
```

13.11 Questões de Segurança

Trataremos do tópico segurança num capítulo à parte (Capítulo 21), porém várias questões de segurança relevantes a redes IP merecem uma discussão já aqui. Nesta seção, veremos brevemente alguns poucos recursos de rede que

27. Se você experimentar este comando na forma **sudo echo 1 > icmp_echo_ignore_broadcasts**, você apenas gerará uma mensagem "permission denied", ou seja, de permissão negada — seu shell tenta abrir o arquivo de saída antes de executar **sudo**. Você quer que **sudo** se aplique tanto ao comando **echo** quanto ao redirecionamento. Portanto, você tem de criar um subshell root no qual possa executar o comando completo.
28. Você tem de instalar o código-fonte do kernel para que esse arquivo exista.
29. No código-fonte do kernel 2.4.22, o arquivo **proc.txt** tem permissões de acesso 0644, o que permite a leitura por qualquer usuário (N.R.T.).

www.snow666.blogspot.com
Snow666

adquiriram má fama por provocarem problemas de segurança e recomendações para minimizar seu impacto. Os detalhes do comportamento-padrão das distribuições Linux vistas neste livro em relação a estes problemas (e métodos apropriados para modificá-los) são vistos posteriormente, nesta seção.

Encaminhamento IP

Se um computador Linux estiver com o encaminhamento IP habilitado, ele poderá agir como um roteador. A menos que seu sistema tenha várias interfaces de rede e realmente deva atuar como um roteador, é recomendável inibir este recurso. Hosts que encaminham pacotes podem, algumas vezes, ser coagidos a comprometer a segurança do sistema fazendo com que pacotes externos pareçam ter sido provenientes de dentro de sua rede. Esse subterfúgio pode ajudar pacotes nocivos a escapar de scanners de rede e de filtros de pacotes.

Redirecionamentos ICMP

Redirecionamentos ICMP podem ser usados maliciosamente para redirecionar tráfego e desorganizar suas tabelas de roteamento. A maioria dos sistemas operacionais os ouve e segue suas instruções como padrão. Seria ruim se todo o seu tráfego de informações fosse redirecionado para a rede de um concorrente por algumas horas, especialmente quando estão sendo feitos backups! Recomendamos que você configure seus roteadores (e hosts que atuam como roteadores) para ignorarem e talvez registrarem em log redirecionamentos ICMP.

Roteamento de origem

O mecanismo de roteamento de origem do IP permite que especifiquemos explicitamente uma série de gateways pelos quais um pacote deve transitar no trajeto a seu destino. O roteamento de origem prevalece sobre o algoritmo de roteamento de próximo salto que é normalmente executado em cada gateway para determinar como um pacote deve ser encaminhado.

O roteamento de origem fazia parte da especificação IP original; ele era destinado a, basicamente, facilitar testes. Ele pode criar problemas de segurança, pois os pacotes normalmente são filtrados conforme suas origens. Se alguém, habilmente, for capaz de direcionar um pacote para parecer que ele se originou dentro de sua rede e não da Internet, ele poderia passar pelo seu firewall. Recomendamos que você não aceite nem encaminhe pacotes roteados pela origem.

Pings transmitidos amplamente e outras formas de difusão ampla direcionada

Os pacotes ping destinados ao endereço de difusão ampla de uma rede (em vez de um endereço de host em particular) normalmente serão entregues a todos os hosts da rede. Tais pacotes têm sido usados em ataques de recusa de serviços; por exemplo, os assim chamados ataques smurf. A maioria dos hosts possui uma maneira de desabilitar pings de difusão ampla — isto é, podem ser configurados para não responderem nem encaminhá-los. O seu roteador Internet também pode filtrar pings de difusão ampla antes de chegarem à sua rede interna. É uma boa idéia usar medidas de segurança tanto em nível de firewall quanto de host caso você possa.

Os pings de difusão ampla são uma espécie de "difusão ampla direcionada" na qual eles são pacotes enviados ao endereço de difusão ampla de uma rede distante. A manipulação-padrão de tais pacotes tem mudado gradualmente. Por exemplo, o IOS da Cisco até a versão 11.x encaminhava pacotes de difusão ampla direcionados como padrão, ao passo que a partir da versão 12.0 isto já não é verdade. Normalmente é possível convencer sua pilha TCP/IP a ignorar pacotes de difusão ampla provenientes de tão longe, mas, já que este comportamento pode ser configurado em cada interface, esta pode ser uma tarefa não trivial em uma instalação grande.

Falsificação de IP (IP spoofing)

O endereço de origem de um pacote IP normalmente é preenchido pelas bibliotecas de rede e é o endereço IP do host do qual o pacote foi enviado. Entretanto, se o software que está criando o pacote usar um soquete bruto, ele poderá preencher qualquer endereço de origem que quiser. Isso é chamado de falsificação de IP e normalmente está associado a algum tipo de comportamento estranho da rede. A máquina identificada pelo endereço IP de origem adulterado (se for um endereço real) em geral é a vítima no esquema. Pacotes retornados e de erros podem corromper ou inundar as conexões de rede da vítima.

Você deve impedir spoofing de IP em seu roteador de fronteira bloqueando pacotes de saída cujos endereços de origem não se encontrem dentro do seu espaço de endereços. Essa precaução é especialmente importante se sua instalação for uma universidade onde os estudantes adoram ficar experimentando e normalmente são vingativos contra "idiotas" em seus canais IRC (Internet Relay Chat) prediletos.

Snow666

Ao mesmo tempo, se você estiver usando internamente espaço de endereços privados, poderá fazer uma filtragem para capturar quaisquer endereços internos que estejam escapando para a Internet. Tais pacotes jamais podem ser respondidos (graças à falta de uma rota de backbone) e normalmente indicam que sua instalação possui algum erro de configuração interno.

Os firewalls baseados em Linux, descritos na próxima seção, fornecem uma maneira de implementar tal filtragem host por host. Entretanto, a maioria das instalações prefere implementar esse tipo de filtragem em seus roteadores de fronteira e não em cada host. Essa é a abordagem que também recomendamos. Descrevemos os firewalls baseados em hosts somente para concluirmos e para situações especiais.

Você também tem de se proteger contra um hacker forjando o endereço de origem em pacotes externos para lograr o seu firewall e fazê-lo pensar que eles se originaram em sua rede interna. O parâmetro de kernel **rp_filter** (configurável no diretório **/proc/sys/net/ipv4/conf/***ifname*) pode ajudá-lo a detectar tais pacotes; **rp** significa reverse path, ou seja, caminho inverso. Se você configurar essa variável em 1, o kernel descartará pacotes que chegam numa interface que seja diferente daquela na qual eles deveriam partir caso o endereço de origem fosse o de destino. Esse comportamento fica ativado na situação-padrão.

Se sua instalação tiver várias conexões com a Internet, pode ser perfeitamente razoável rotas de chegada e de saída serem diferentes. Nesse caso, você terá de configurar **rp_filter** em 0 para fazer com que o seu protocolo de roteamento funcione perfeitamente. Se sua instalação tiver somente uma saída para a Internet, então configurar **rp_filter** em 1 normalmente é seguro e apropriado.

Firewalls baseados em hosts

O Linux inclui software para filtragem de pacotes (também conhecido como firewall). Apesar de descrevermos este software posteriormente neste capítulo e também no Capítulo 21, *Segurança*, realmente não recomendamos usar uma estação de trabalho como um firewall. A segurança do Linux (especialmente aquela fornecida pelos nossos amigáveis fornecedores) é fraca e a segurança do NT é pior ainda. Sugerimos que você adquira uma solução de hardware dedicada para usar como firewall. Até mesmo uma solução de software sofisticada como o produto Firewall-1 da Checkpoint (que roda num host Solaris) não é tão bom quanto um hardware dedicado como o equipamento PIX da Cisco — e seu preço é praticamente o mesmo!

Uma discussão mais aprofundada sobres questões relacionadas a firewalls está no Capítulo 21.

Redes Privadas Virtuais - VPNs

Muitas empresas que têm escritórios em várias partes do mundo gostariam que todas estas localidades se conectassem a uma grande rede privada. Infelizmente, o custo de arrendar uma linha de dados transoceânica ou até mesmo transcontinental pode se tornar proibitivo. Tais empresas podem, na verdade, usar a Internet como se fosse uma linha de dados privada estabelecendo uma série de "túneis" criptografados e seguros entre suas diversas localidades. Uma rede "privada" que inclui tais túneis é conhecida como VPN (Virtual Private Network, rede privada virtual).

Consulte a seção 21.12, para obter maiores informações sobre o IPSEC.

Algumas VPNs usam o protocolo IPSEC, que foi padronizado pelo IETF em 1998. Outras usam soluções proprietárias que normalmente não interoperam umas com as outras. Caso precise de funcionalidade VPN, sugerimos que dê uma olhada em produtos como o roteador 3660 da Cisco ou o Watchguard Firebox, ambos capazes de fazer tunelamento e criptografia. O dispositivo Watchguard usa PPP para uma porta serial para fins de gerenciamento. Um administrador de sistemas pode discar para o dispositivo para configurá-lo ou para acessar a VPN para testes.

Para uma solução VPN para quem está com o orçamento curto, veja o exemplo do Capítulo 12 que usa PPP sobre uma conexão **ssh** para implementar uma rede privada virtual.

Variáveis de kernel relacionadas com segurança

A Tabela 13.17 ilustra o comportamento-padrão do Linux em relação a diversas questões sensíveis de rede. Para uma breve descrição das implicações desses comportamentos, consulte a seção 13.11. Recomendamos que você altere os valores dessas variáveis para que não responda a pings de difusão ampla, não dê ouvidos a redirecionamentos de rotas e não aceite pacotes roteados pela origem.

Snow666

Tabela 13.17 Comportamentos de rede relacionados com segurança no Linux.

Recurso	Host	Gateway	Arquivo de controle (em /proc/sys/net)
Encaminhamento IP	inativo	ativo	**ipv4/ip_forward** para todo o sistema
			ipv4/conf/ *interface*/**forwarding** por interface[a]
Redirecionamentos ICMP	obedece	ignora	**ipv4/conf/** *interface*/**accept_redirects**
Roteamento de origem	ignora	obedece	**ipv4/conf/** *interface*/**accept_source_route**
Ping de difusão ampla	responde	responde	**ipv4/icmp_echo_ignore_broadcasts**

a. *interface* pode ser um nome de interface específico ou então **all**.

Para alterar qualquer um desses parâmetros de maneira permanente (ou mais precisamente, para reiniciá-los cada vez que o sistema for inicializado), acrescente os comandos **echo** apropriados a um script que é executado durante a seqüência de boot ou use o comando **sysctl**.

No Red Hat normalmente fazemos isso adicionando valores a **/etc/sysctl.conf**, que é lido pelo comando **sysctl** no momento do boot (por solicitação do script de inicialização **network**). O formato de **sysctl.conf** é *variável=valor*, e não **echo value > variable** como você faria para alterar as coisas manualmente. Os nomes de variáveis são nomes de caminhos relativos a **/proc/sys**; você também pode usar pontos em vez de barras caso prefira. Por exemplo, qualquer uma das linhas abaixo

```
net.ipv4.ip_forward=0
net/ipv4/ip_forward=0
```

no arquivo **/etc/sysctl.conf** faria com que o encaminhamento IP fosse desativado (para este host). Você também pode controlar o encaminhamento em **/etc/sysconfig/network**.

A distribuição SuSE inclui o comando **sysctl,** porém não executa o comando em seus scripts de boot e não vem com um arquivo **/etc/sysctl.conf** bem comentado como aqueles com os quais nos acostumamos. Para alterar os valores das variáveis de kernel relevantes à rede, você pode simular a manipulação da variável IP_FORWARD (por meio) de **rc.config** (em **/etc/init.d/boot**) ou acrescentar a sua própria chamada ao comando **sysctl** em algum ponto na seqüência de inicialização.

O Debian fornece efetivamente um modelo de arquivo **sysctl.conf** e também chama **sysctl** com este arquivo durante a inicialização. Um dos exemplos fornecidos configura o kernel para ignorar pings de difusão ampla, conforme sugerimos anteriormente.

13.12 NAT NO LINUX (MASCARAMENTO DE IP)

O Linux tradicionalmente implementa somente uma forma limitada do NAT (Network Address Translation, tradução de endereços de rede) que é mais apropriadamente chamado de PAT (Port Address Translation, tradução de endereços de portas). Em vez de usar um intervalo de endereços IP como uma verdadeira implementação NAT faria, o PAT multiplexa todas as conexões num único endereço. Para gerar ainda mais confusão, documentos do Linux freqüentemente se referem ao recurso nem como NAT nem como PAT, mas sim como "mascaramento de IP". Os detalhes e diferenças não são de muita importância prática, de modo que iremos nos referir à implementação Linux como NAT para fins de consistência.[30]

O Linux inclui alguns programas bem razoáveis para configurar o NAT. O software predominante na época em que este livro foi escrito era **ipchains**, que todas as distribuições usadas como exemplo em nosso livro fornecem. Um novo e aperfeiçoado software para filtragem de pacotes de espaço de usuário chamado **iptables** usa o recurso "netfilter" no kernel 2.4 do Linux e se encontra na versões atuais do Red Hat, Debian e SuSE. Esperamos que **iptables** substitua completamente **ipchains** num prazo relativamente curto. Em sistemas que possuem ambos, você pode usar somente um dos sistemas de cada vez; sugerimos o **iptables**.

Ambos os pacotes implementam não somente o NAT como também a filtragem de pacotes. Em **ipchains** isto era um pouco confuso, porém **iptables** faz uma distinção muito mais clara entre os recursos NAT e de filtragem. Os recursos de filtragem de pacotes são vistos de maneira mais detalhada no Capítulo 21, *Segurança*, começando na página 504; aqui, restringimos nossa discussão a alguns exemplos de configuração do NAT.

30. O software mais recente **iptables** presente no kernel versão 2.4 pode, de fato, fornecer NAT verdadeiro, porém ainda resta ver se este recurso será adotado amplamente.

www.snow666.blogspot.com
Snow666

Para que o mascaramento IP funcione, você tem de habilitar o encaminhamento IP e construir o kernel com CONFIG_IP_MASQUERADE definido. Também ajuda configurar a variável de kernel ip_masq_debug para facilitar a depuração (veja o comando **sysctl** ou altere o conteúdo de **/proc/sys/net/ipv4/ip_masq_debug** para 1).

Por exemplo, para configurar o mascaramento IP (também conhecido como NAT ou PAT) para dissimular o espaço de endereços privado usado na rede interna 192.168.1.0/24, você poderia usar o seguinte comando:

```
ipchains -A forward -i ppp0 -s 192.168.1.0/24 -d ! 192.168.1.0/24 -j MASQ
```

Nesse caso, especificamos os critérios de concordância em que o endereço de origem deve ser interno e o endereço de destino externo (o **!** nega o sentido do teste) para ocorrer o mascaramento IP. Tráfego interno que por acaso passe através deste host não será afetado.

Já que **ipchains** implementa o PAT em vez do verdadeiro NAT, não há nenhuma necessidade de especificar um intervalo de endereços externos a serem usados na Internet. O gateway Linux usa o seu próprio endereço IP para todo o tráfego externo e utiliza números de portas para multiplexar conexões de vários hosts internos.

O comando **iptables** para associar pacotes da rede 192.168.1.0/24 para um intervalo de 10 endereços na rede roteável 128.138.198.0 seria:

```
iptables -A POSTROUT NG SNAT --to-source 128.138.198.1-128.138.198.10
```

Também podemos especificar um intervalo de portas de origem anexando: *startport-stopport* ao comando. Em geral, é melhor deixar **iptables** gerenciar as portas por conta própria. Como padrão, **iptables** não irá modificar a porta de origem a menos que seja necessário para fins de exclusividade.

13.13 PPP: O Protocolo Ponto-a-Ponto

PPP (Point-to-Point Protocol, protocolo ponto-a-ponto) tem a distinção de ser usado tanto nos links Internet mais lentos quanto naqueles mais velozes. Em sua forma síncrona, ele é o protocolo de encapsulamento usado em circuitos de alta velocidade que possuem roteadores robustos em ambas as pontas. Em sua forma assíncrona, ele é um protocolo de encapsulamento de linhas seriais que especifica como os pacotes IP têm de ser codificados para transmissão numa linha serial lenta (e normalmente não confiável). As linhas seriais apenas transmitem fluxos de bits e não têm noção alguma do ponto de início ou final de um pacote. O driver de dispositivo PPP cuida da codificação e decodificação de pacotes na linha serial; ele acrescenta um cabeçalho em nível de link e marcadores que separam os pacotes.

O PPP é algumas vezes usado com as tecnologias domésticas mais modernas como DSL e modems a cabo, porém este fato normalmente é omitido de um administrador de sistemas. O encapsulamento é executado tipicamente pelo dispositivo de interface e o tráfego é transferido via Ethernet. Você vê apenas uma conexão Ethernet.

Projetado por comitê, o PPP é o protocolo de encapsulamento "tudo o que se possa imaginar *e* mais um pouco". Ele se inspirou nos protocolos SLIP (Serial Line IP) e CSLIP (Compressed SLIP) desenvolvidos por, respectivamente, Rick Adams e Van Jacobson. O PPP difere desses sistemas pelo fato de permitir a transmissão de vários protocolos através de um único link. Ele é especificado na RFC1331.

Resolvendo problemas de desempenho no PPP

O PPP fornece toda a funcionalidade do padrão Ethernet, porém em velocidades *muito* menores. As LANs empresariais comuns operam até 10 Mb/s ou 100 Mb/s — isto é, de 10.000 Kb/s a 100.000 Kb/s. Uma conexão discada opera a aproximadamente de 28 Kb/s a 56 Kb/s.[31] Para compreender melhor esses números, leva aproximadamente 5 minutos para transferirmos um arquivo de um megabyte através de uma rede PPP discada. A velocidade é aceitável para e-mail ou navegação Web com imagens desativadas, porém sites Web muito empetecados o levarão à loucura. Para melhorar o desempenho interativo, você poderá configurar o MTU do ponto-a-ponto para um valor bem baixo. Ele normalmente adota como padrão 512 bytes; tente 128 caso esteja realizando muitas tarefas interativas. Se estiver usando PPP através de Ethernet, use **tcpdump** para ver os tamanhos dos pacotes trafegando pela rede e configure o MTU de acordo. O MTU de redes Ethernet é de 1500, porém o encapsulamento PPP torna valores ligeiramente menores mais eficientes. Por exemplo, **pppoe** sugere 1412 bytes para hosts por trás da conexão PPP e 1492 no link PPP. Certamente você não quer que cada pacote seja fragmentado porque você configurou o seu MTU padrão com um valor muito alto.

---

31. O PPP é normalmente usado a 19.200 bps. Tecnicamente, ele pode ser usado em links mais lentos, porém se torna terrivelmente mais lento.

Snow666

Veja o Capítulo 17 para obter maiores informações sobre o NFS.

Executar NFS através de um link PPP pode ser terrivelmente lento. Você deve considerá-lo somente se puder rodar NFS via TCP em vez de UDP.

O protocolo X Windows usa TCP, de modo que é possível executar aplicativos X através de um link PPP. Programas como **xterm** funcionam bem, contudo evite aplicativos que usem fontes extravagantes ou imagens bitmap.

Conectando-se a uma rede via PPP

Para conectarmos um host a uma rede via PPP, temos de atender a três pré-requisitos:

- O kernel do host deve ser capaz de enviar pacotes IP através de uma linha serial conforme especificado pelo padrão de protocolo PPP.
- Temos de ter um programa em nível de usuário que nos permita estabelecer e manter conexões PPP.
- Um host no outro lado da linha serial tem de entender o protocolo que estamos usando.

*Veja a seção 13.7 para obter maiores informações sobre **ifconfig**.*

Fazendo um host falar PPP

Para estabelecer uma conexão PPP, o host tem de ser capaz de enviar e receber pacotes PPP. Em sistemas Linux, o PPP é um módulo de kernel carregável que coloca pacotes de rede na fila de saída de dispositivo serial e vice-versa. Esse módulo normalmente simula ser mais uma interface de rede, para poder ser manipulado com ferramentas de configuração padrão como **ifconfig**.

Controlando links PPP

A seqüência exata dos eventos envolvidos no estabelecimento de uma conexão PPP depende do sistema operacional e do tipo de servidor para o qual estamos discando. As conexões podem ser iniciadas tanto manual quanto dinamicamente.

Para estabelecer uma conexão PPP manualmemte, executamos um comando que disca um modem, faz o login num host remoto e inicia o mecanismo de protocolo PPP remoto. Se esse procedimento for bem-sucedido, a porta serial é então configurada como uma interface de rede. Essa opção normalmente deixa o link ativo por um tempo longo, o que a torna mais indicada para uma linha telefônica dedicada à conectividade IP.

Numa configuração dinâmica, um daemon observa suas interfaces de "rede" seriais para ver quando o tráfego foi enfileirado para elas. Quando alguém tenta enviar um pacote, o daemon disca um modem automaticamente para estabelecer a conexão, transmite o pacote e, se a linha retorna ao estado de ocioso, desconecta a linha após um período de tempo razoável. A discagem dinâmica é normalmente usada caso uma linha telefônica transporte tanto dados quanto voz ou se a conexão envolve tarifas de tempo de conexão e de chamadas interurbanas.

O software para implementar esses dois esquemas de conexão é incluso na maioria das versões do PPP.

Localizando um host para dialogar

Se você estiver configurando um link entre duas instalações dentro de sua empresa ou, quem sabe, entre sua casa e o escritório, basta instalar o software PPP em ambas as extremidades. Entretanto, caso pretenda usar o PPP para obter uma conexão Internet, provavelmente você precisará tratar com um ISP comercial. A maioria dos ISPs oferece serviço PPP discado para o público a um preço razoável. Muitos também suportam conexões DSL ou via modem a cabo que usam o PPP através de mídia mais rápida.

Atribuindo um endereço

Da mesma forma que atribuímos um endereço IP a um novo host numa rede Ethernet, precisamos atribuir um endereço IP a cada interface PPP. Existem várias maneiras de atribuir endereços a estes links (inclusive não atribuir nenhum endereço). Discutimos somente o método mais simples aqui.

Imagine um enlace PPP como uma rede de sua propriedade. Isto é, uma rede formada exatamente por dois hosts, freqüentemente denominada de rede ponto-a-ponto. Você precisa atribuir um número de rede ao link exatamente como faria para atribuir um número de rede a um novo segmento numa rede Ethernet, usando seja lá quais forem as regras em vigor em sua instalação. Você pode escolher dois endereços de host quaisquer dessa rede e atribuir um a cada extremidade do link. Siga também outros costumes locais como, por exemplo, padrões de sub-redes. Cada host se torna então um "gateway" para a rede ponto-a-ponto pelo menos no que se refere ao restante do mundo. (No mundo real, normalmente não controlamos ambas as pontas do link; nosso ISP nos dá o endereço IP que devemos usar na outra ponta.)

O DHCP também pode ser usado para atribuir o endereço IP à extremidade de um link PPP. Alguns ISPs oferecem serviço doméstico que usa DHCP e serviço comercial que custa mais, porém inclui endereços estáticos.

Snow666

Veja o Capítulo 14 para obter maiores informações sobre roteamento.

Roteamento

Já que o PPP requer que o servidor remoto aja como um roteador IP, precisamos nos preocupar com o roteamento IP exatamente como faríamos com um gateway "real", tal como uma máquina que conecta duas redes Ethernet. O objetivo do roteamento é o de direcionar pacotes através de gateways de modo que eles possam atingir os seus destinos finais. O roteamento pode ser configurado de diversas maneiras.

Uma conexão host-cliente PPP comum deve ter uma rota-padrão que encaminha pacotes ao servidor PPP. Da mesma maneira, o servidor precisa ser conhecido dos demais hosts em sua rede como o gateway para a máquina na extremidade (folha).

A maioria dos pacotes PPP cuida dessas tarefas de roteamento de modo automático.

Veja o Capítulo 21 para obter maiores informações sobre segurança.

Garantindo a segurança

Toda vez que você adiciona um host a uma rede surgem problemas de segurança. Já que um host conectado via PPP é um membro fidedigno da rede, precisamos tratá-lo como tal: verificar se o sistema não tem contas sem senhas ou com senhas inseguras, que todas as correções de segurança dos diversos fornecedores tenham sido instalados e assim por diante. PPP no Linux suporta dois protocolos de autenticação: PAP (Password Authentication Protocol) e CHAP (Challenge Handshake Authentication Protocol).

Usando servidores de terminais

Talvez você ache que por começar a oferecer conexões PPP a usuários domésticos, tenha mais solicitações do que portas seriais. Uma série de servidores de terminais oferece recursos de PPP, e servidores recentes também possuem modems integrados. Nosso favorito é o Lucent (antigamente Livingston) Portmaster 3. A série Cisco Access Server AS5x00 também é muito popular. Esses produtos fornecem uma fonte conveniente e de fácil manutenção de portas seriais completas com software PPP já instalados. Elas permitem que estabeleçamos um "pool" de modems dial-in que oferecem serviço PPP para usuários fora do site.

Usando "chat scripts"

A implementação PPP de linha serial do Linux usa um "chat script" para conversar com o modem e também para fazer o login numa máquina remota e inicializar um servidor PPP. Um chat script é formado por uma seqüência de strings para enviar e strings esperadas em retorno, com uma forma limitada de instrução condicional que pode expressar conceitos como "espere receber a string 'Login', mas se você não a receber, envie um carriage return e aguarde por ela novamente".

A idéia de um chat script originou-se com o sistema armazene-e-encaminhe UUCP de dias já idos. Nos anos 80, as máquinas chamavam uma a outra no meio da noite, faziam o login através de chat scripts e trocavam arquivos entre si. Apesar da reivindicação popular, o UUCP ainda não está completamente morto: o usuário UUCP é o proprietário do grupo de arquivos de dispositivos seriais no Linux SuSE, e é preciso ser um membro do grupo UUCP para usar um modem dial-out para PPP.

A maioria das implementações PPP vem com modelos de chat scripts que podemos adaptar ao nosso ambiente. Precisaremos editar os scripts para configurar parâmetros como o número de telefone a ser chamado e o comando a ser executado após um login bem-sucedido. A maioria dos chat scripts contém uma senha em texto não criptografado; configure as permissões apropriadamente.

Configuração PPP no Linux

Os modems (juntamente com as impressoras) sempre foram uma pedra no sapato dos administradores de sistemas. E não é nenhuma surpresa, quando o software para configurar uma conexão PPP através de um modem aleatório possui mais de 125 opções possíveis — de longe um número excessivo para se avaliar e configurar cuidadosamente.

O Red Hat, o SuSE e o Debian vêm com o pacote PPP de Paul Mackerras que usa um daemon chamado **pppd** e mantém a maioria de seus arquivos de configuração em **/etc/ppp**. Execute o comando **pppd --version** para ver qual versão do pacote PPP foi instalada em sua distribuição em particular.

O Red Hat, o SuSE e o Debian incluem uma versão do PPP da Roaring Penguin Software que é projetada para uso através de redes Ethernet (por exemplo, numa conexão DSL para um ISP local). Todas as três distribuições adotadas como exemplo neste livro incluem também suporte PPP para conexões ISDN. Os arquivos de configuração para essas mídias adicionais são colocados com aquelas para PPP via links seriais no diretório **/etc/ppp**. Os nomes

Snow666

dos arquivos normalmente são similares, porém com o acréscimo de um **oe** de "over Ethernet" ou **i** de ISDN. A Tabela 13.18 mostra as localizações dos comandos e arquivos de configuração relevantes.

Veja a seção 7.6 para maiores informações sobre os nomes de portas seriais.

Em nossos exemplos de arquivos de configuração, chamamos a porta serial que possui um modem conectado a ela **/dev/modem**. Algumas distribuições na verdade possuem um arquivo **/dev/modem** que é um link para uma das portas seriais do sistema (normalmente **/dev/ttyS0** ou **/dev/ttyS1**), porém esta prática é abandonada atualmente. Você deve substituir o arquivo de dispositivo apropriado para o seu caso em particular. O SuSE define de forma útil uma variável chamada MODEM em **/etc/rc.config** para uso em seus scripts de inicialização, contudo parece não usá-la na prática.

Além do software PPP, cada distribuição fornece vários programas que na verdade discam o telefone para estabelecer uma conexão. Os exemplos a seguir usam **pppd** com chat scripts e **wvdial**, que é incluso em todas as distribuições aqui apresentadas. Não descrevemos as alternativas para **wvdial**, porém a Tabela 13.19 mostra onde procurar e o que procurar caso você esteja interessado em comparar.

Tabela 13.18 Arquivos de configuração e comandos relacionados ao PPP classificados.

Sistema	Comandos ou arquivos de configuração	Descrição
Red Hat, SuSE, Debian (serial)	**/usr/sbin/pppd**	Programa daemon PPP
	/usr/sbin/chat	Conversa com o modem
	/usr/sbin/pppstats	Mostra dados estatísticos do link PPP
	/usr/sbin/pppdump	Transforma pacotes PPP em ASCII legível
	/etc/ppp/options	Arquivo de configuração para **pppd**
Debian (extras seriais)	**/usr/bin/pon**	Inicia uma conexão PPP
	/usr/bin/poff	Encerra uma conexão PPP
	/usr/bin/plog	Mostra o final de **ppp.log**
	/usr/sbin/pppconfig	Configura **pppd**
	/etc/ppp/peers/provider	Opções para **pon** para contactar seu ISP
	/etc/chatscripts/provider	Chat script para **pon** para dialogar com o ISP
Red Hat (DSL)	**/usr/sbin/pppoe**	Cliente PPP em rede Ethernet
	/usr/sbin/pppoe-server	Servidor PPP em rede Ethernet
	/usr/sbin/pppoe-sniff	Farejador que depura peculiaridades do provedor
	/usr/sbin/adsl-connect	Script que gerencia links
	/usr/sbin/adsl-setup	Script que configura **pppoe**
	/usr/sbin/adsl-start	Script que aciona link **pppoe**
	/usr/sbin/adsl-stop	Script que desativa link **pppoe**
	/usr/sbin/adsl-status	Mostra o estado do link **pppoe**
	/etc/ppp/pppoe.conf	Arquivo de configuração usado por **adsl-***
	/etc/ppp/pppoe-server-options	Arquivo com opções extras para servidor
SuSE (DSL)	**/usr/sbin/pppoed**	PPP através de cliente Ethernet
	/etc/pppoed.conf	Arquivo de configuração para **pppoed**
Debian (DSL)	**/usr/sbin/pppoe**	Cliente ppp em rede Ethernet
	/usr/sbin/pppoe-server	Servidor ppp em rede Ethernet
	/usr/sbin/pppoe-relay	Agente de repasse ppp em rede Ethernet
	/usr/sbin/pppoe-sniff	Examina a rede para frames pppoe não padrão
	/etc/ppp/peers/dsl-provider	Arquivo de configuração para ppp em rede Ethernet
	/etc/ppp/ppp_on_boot.dsl	Para estabelecer a conexão na inicialização.[32]

32. Quando se deseja que a conexão seja estabelecida automaticamente na inicialização, deve-se criar um link simbólico para o arquivo ppp_on_boot.dsl chamado **ppp_on_boot**. Isso pode ser feito com o seguinte comando em um shell como root: **cd/etc/ppp**; **ln -s ppp_on_boot.dsl ppp_on_boot\line**. Para iniciar a conexão **ppp** execute **/etc/init.d/ppp start**. A conexão também será iniciada automaticamente na próxima inicialização (N.R.T.).

Snow666

Tabela 13.18 Arquivos de configuração e comandos relacionados ao PPP classificados. (*continuação*)

Sistema	Comandos ou arquivos de configuração	Descrição
Todos (ISDN)	**/usr/sbin/ipppd**	PPP através de daemon ISDN
	/usr/sbin/ipppstats	Mostra dados estatísticos PPP ISDP
	/etc/ppp/ioptions	Opções para **ipppd**

Tabela 13.19 Discadores PPP classificados por sistema.

Sistema	Discador	Exemplos de configuração
Red Hat	**wvdial**	–
	dip	–
SuSE	**wvdial**	Exemplo de arquivo **/etc/wvdial.conf** bem comentado
	diald	Exemplo de arquivo **/etc/diald.conf** e **/usr/lib/diald/***
	dip	Exemplo de arquivo **/etc/diphosts**
Debian	**wvdial**	Inclui exemplos de configurações
	diald	Exemplos de configuração em **/etc/diald**

Agora que mencionamos as portas de modem e o software discador, vamos falar sobre como configurar **pppd** para usá-los. As opções globais são configuradas no arquivo **/etc/ppp/options** e opções para conexões particulares podem ser armazenadas nos diretórios **/etc/ppp/peers** e **/etc/chatscripts** (no Debian). O Red Hat e o SuSE tendem a colocar chat scripts no diretório **/etc/ppp** com nomes como **chat.***remotehost*. Alternativamente, no Red Hat, o arquivo **/etc/sysconfig/network-scripts/ifcfg-***ttyname* pode incluir opções específicas de conexões para uma determinada interface PPP.

Como padrão, **pppd** consulta primeiramente o arquivo **options**, depois o arquivo de inicialização pessoal do usuário, **~/.ppprc**, a seguir o arquivo específico a uma determinada conexão **options.***ttyname* (se existir um) e, finalmente, seus argumentos de linha de comando.

Um truque prático sugerido por Jonathan Corbet, um velho marujo do Linux, é o de definir mais de uma interface PPP: uma para casa, uma para hotéis enquanto estiver viajando etc. Essa configuração facilita a troca de contextos.

wvdial é mais inteligente do que **chat** e possui um comportamento-padrão sensível caso sejam deixados parâmetros sem especificação. **wvdial** obtém suas informações de configuração de **/etc/wvdial.conf**: detalhes de modems, nome de login, senha, número de telefone etc. Podemos fornecer informações para vários destinos num único arquivo de configuração. Use o programa **wvdialconf** para descobrir as características de seu modem e crie um arquivo **wvdial.conf** inicial para ele.

Os arquivos de configuração abaixo são extraídos de várias configurações PPP diferentes. O primeiro arquivo, **/etc/ppp/options**, configura opções globais para **pppd**. As opções ativas para cada distribuição conforme saem de fábrica são mostradas a seguir:

Red Hat: **/etc/ppp/options**

```
lock
```

SuSE: **/etc/ppp/options**

```
debug
noauth
crtscts
lock
modem
asyncmap 0
nodetach
lcp-echo-interval 30
lcp-echo-failure 4
idle 600
noipx
```

Snow666

Debian: **/etc/ppp/options**

```
asyncmap 0
auth
crtscts
lock
hide-password
modem
proxyarp
lcp-echo-interval 30
lcp-echo-failure 4
noipx
```

Na Universidade de Colorado, gostamos de usar o seguinte arquivo de opções:

```
# Global PPP options
lock                    # Opões Globais do PPP
asyncmap 0x00000000     # Sempre bloqueia o dispositivo que você está utilizando
crtscts                 # Por padrão não protege (escape) nenhum caracter
defaultroute            # Usa o controle de fluxo de hardware
mru 552                 # Adiciona uma rota padrão através da interface ppp
mtu 552                 # MRU/MTU 512 (dados) + 40 (cabeçalho)
```

A seguir, um modelo de arquivo **/etc/sysconfig/network-scripts/ifcgf-ppp0** de um sistema Red Hat. Este arquivo-esqueleto foi construído pelo utilitário **linuxconf**.

```
PERSIST=yes
DEFROUTE=yes
ONBOOT=no
INITSTRING=ATZ
MODEMPORT=/dev/modem
LINESPEED=115200
ESCAPECHARS=no
DEFABORT=yes
HARDFLOWCTL=yes
DEVICE=ppp0
PPPOPTIONS=
DEBUG=yes
PAPNAME=remote
REMIP=
IPADDR=
BOOTPROTO=none
MTU=
MRU=
DISCONNECTTIMEOUT=
RETRYTIMEOUT=
USERCTL=no
```

Eis um exemplo de chat script (**chat-ppp0**) correspondente ao arquivo **ifcfg-ppp0** anterior (com toda a sua sintaxe concisa e ligeiramente bizarra):

```
'ABORT' 'BUSY'
'ABORT' 'ERROR'
'ABORT' 'NO CARRIER'
'ABORT' 'NO DIALTONE'
'ABORT' 'Invalid Login'
'ABORT' 'Login incorrect'
' ' 'ATZ'
'OK' 'ATDT phone-number'
'CONNECT' ' '
'TIMEOUT' '120'
'ogin:' 'account'
'ord:' 'password'
'TIMEOUT' '5'
'~--' ' '
```

Sn❁w666

Várias linhas desse chat script contêm um parâmetro nulo indicado por um par de aspas simples, que se parecem com aspas duplas nesta fonte.

Normalmente podemos adaptar um chat script existente ao nosso ambiente sem nos preocuparmos muito sobre o seu funcionamento exato. Aqui, as linhas iniciais configuram algumas condições gerais nas quais o script deve abortar. As linhas seguintes inicializam o modem e discam o telefone, e as linhas restantes esperam por uma conexão e introduzem o nome de usuário e senha apropriados.

O timeout no chat script algumas vezes precisa ser ajustado para lidar com situações de discagem mais complicadas como aquelas em hotéis ou estabelecimentos comerciais com comutadores telefônicos locais ou para lidar com o sinal de voice mail que algumas empresas usam antes de darem um tom de discagem real. Na maioria dos modems, uma vírgula no número de telefone indica uma pausa na discagem. Talvez sejam necessárias várias vírgulas caso você tenha de discar um determinado dígito e depois aguardar um segundo tom de discagem antes de prosseguir.

Logins PPP em nossa instalação são apenas nomes de usuários com um P na frente deles. Essa convenção facilita memorizar a quem uma determinada máquina PPP pertence.

A associação entre **ifcfg-ppp0** e **chat.ppp0** é feita pelo comando **ifup**, que será executado automaticamente durante a inicialização caso o arquivo **ifcfg** exista. Também podemos chamar **pppd** explicitamente com um arquivo de opções específicas a uma determinada conexão como um argumento, desde que o arquivo contenha uma linha `connect` que liste o nome de arquivo chat correspondente.

Nosso próximo exemplo de discagem é de um sistema Debian. Ele usa o diretório **peers**, coloca seu chat script no diretório **/etc/chatscripts** e usa o mecanismo de autenticação PAP em vez de armazenar a senha no chat script. Primeiramente, as opções para esta conexão, **/etc/ppp/peers/my-isp**:

```
/dev/modem          ### preenche a porta serial de seu modem
debug
crtscts
name username       ### nome de usuário em my-isp
remotename my-isp
noauth
noipdefault
defaultroute
connect '/usr/sbin/chat -v -f /etc/chatscripts/my-isp'
```

/etc/chatscripts/my-isp contém as seguintes entradas:

```
'ABORT' 'BUSY'
'ABORT' 'ERROR'
'ABORT' 'NO CARRIER'
'ABORT' 'NO DIALTONE'
' ' 'ATZ'
'OK' 'ATDT phonenumber'
'CONNECT' ' '
'TIMEOUT' 15
'---' ' '
```

O arquivo de autenticação usado para se conectar ao ISP, **/etc/ppp/pap-secrets**, precisa conter a seguinte linha:

```
login-name my-isp password
```

onde *my-isp* é o valor da variável `remotename` nas opções acima. Para ativar a conexão nessa situação, use o comando **pppd call my-isp**.

Eis um exemplo que usa PPP via conectividade Internet genérica existente, mas que se associa a **ssh** para criar uma conexão segura por meio de uma VPN (Virtual Private Network). Mostramos tanto as configurações servidor quanto cliente.

O arquivo **/etc/ppp/options** do servidor:

```
noauth
logfile pppd.log
passive
silent
nodetach
```

Snow666

Cada conexão também possui um arquivo **/etc/ppp/options.***ttyname* que contém as atribuições de endereços IP para a conexão:

```
endereçoIP-local:endereçoIP-remoto
```

O shell do usuário PPP é configurado para **/usr/sbin/pppd** no servidor de modo que o daemon de servidor é iniciado automaticamente. Todas as chaves de autenticação têm de ser configuradas antecipadamente com **ssh-agent** para que não seja solicitada nenhuma senha. No lado cliente, a configuração é feita no diretório **/etc/ppp/peers** com um arquivo nomeado para o servidor — chamemos a configuração de "my-work". O arquivo **/etc/ppp/peers/my-work** do cliente conteria:

```
noauth
debug
logfile pppd.log
passive
silent
pty "ssh -t user@remotehost"
```

Para logar ao escritório a partir de casa numa conexão PPP segura, o usuário teria apenas de digitar **pppd call my-work**.

Finalmente, incluímos um exemplo que usa o comando **wvdial** e sua fácil configuração para evitar toda aquela mágica do chat script que parece ser necessária:

/etc/wvdial.conf:

```
[Dialer Defaults]
Phone = numerodotelefone
Username = nome-de-login
Password = senha
Modem = /dev/ttyS1

[Dialer creditcard]
Phone = codigo-acesso-longa-distancia,,,numerodotelefone,,numero-do-cartao-de-credito
```

Se **wvdial** for chamado sem o emprego de qualquer argumento, ele usará a seção-padrão do discador do arquivo **/etc/wvdial.conf** em **~/.wvdialrc** para fazer a chamada e iniciar o PPP. Se for chamado com um parâmetro (por exemplo, **wvdial creditcard**), ele usará a seção apropriada do arquivo de configuração para prevalecer sobre quaisquer parâmetros que estejam especificados na seção-padrão.

Para interromper uma conexão PPP, é melhor usar **ifdown** do que simplesmente interromper o daemon **pppd**. Se interrompermos **pppd** diretamente, o Linux perceberá e o reiniciará.

```
% sudo ifdown ppp0
```

Se sua máquina for portátil e algumas vezes usa Ethernet em vez de PPP, pode ser que haja uma rota-padrão através da interface Ethernet antes de **pppd** ser inicializado. Infelizmente **pppd** é muito educado para "arrancar" a rota e instalá-la por conta própria, que é o comportamento que realmente queremos. Para solucionar o problema, elimine a rota-padrão existente antes de iniciar **pppd**. É mais fácil você mesmo criar um pequeno script **ppp.up** que elimina a rota-padrão automaticamente.

Eis o aspecto da tabela de roteamento e da configuração da interface PPP depois de a conexão PPP ter sido estabelecida:

```
% ifconfig ppp0
ppp0  Link encap:Point-to-Point Protocol
      inet addr:10.0.0.56 P-t-P:10.0.0.55 Mask:255.255.255.255
      UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1
      RX packets:125 errors:0 dropped:0 overruns:0 frame:0
      TX packets:214 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:3
      RX bytes:11446 (11.1 Kb) TX bytes:105586 (103.1 Kb)

% netstat -nr
Kernel IP routing table
Destination  Gateway      Genmask          Flags  MSS  Window  irtt Iface
10.0.0.55    0.0.0.0      255.255.255.255  UH      40  0          0 ppp0
0.0.0.0      10.0.0.55    0.0.0.0          UG      40  0          0 ppp0
```

Snow666

Podemos obter dados estatísticos sobre a conexão PPP e os pacotes que ele transferiu por meio do comando **pppstats**:

```
% pppstats
   IN PACK VJCOMP VJUNC VJERR |    OUT PACK VJCOMP VJUNC NON-VJ
11862  133      8    96     0 | 110446  226     27    89    110
```

A coluna VJCOMP conta pacotes que usam a compactação de cabeçalho TCP de Van Jacobson, e a coluna VJUNC conta aqueles que não a usam. Veja a RFC1144 para maiores detalhes.

Depurar uma conexão PPP pode realmente ser uma dor de cabeça, pois há muitas peças em jogo. **pppd** submete entradas de log ao recurso daemon por meio de syslog em sistemas Red Hat e Debian e para o recurso local2 no SuSE. Podemos aumentar o nível de registro usando o flag **debug** na linha de comando de **pppd** ou então solicitando mais registro no arquivo de opções. **pppd** também fornece códigos de saída detalhados em caso de falha, de forma que se você tentar executar **pppd** e ele se recusar, execute **echo $status** (antes de fazer qualquer outra coisa) para recuperar o código de saída e depois pesquisar este valor na página de manual do **pppd**.

O Linux SuSE tende a incluir modelos de arquivos de configuração para cada subsistema que são em grande parte comentários, mas que explicam o formato e o significado das opções disponíveis. Os arquivos no diretório **/etc/ppp** do SuSE não são exceção; eles são bem-documentados e contêm valores sugeridos razoáveis para diversos parâmetros.

O Debian também possui modelos de arquivo de configuração bem documentados para PPP. Ele possui um subdiretório dedicado a chat scripts, **/etc/chatscripts**. Para ativar uma interface com PPP, podemos incluí-la no arquivo **/etc/network/interfaces** por meio do método `ppp` e da opção `provider` para amarrar o nome de seu provedor (em nosso caso, my-isp) a um nome de arquivo no diretório **/etc/peers** (**/etc/peers/my-isp**). Por exemplo:

```
iface eth0 inet ppp
   provider my-isp
```

Nesse caso, os comandos **pon** e **poff** específicos do Debian serão usados para controlar a conexão.

13.14 Peculiaridades de Rede do Linux

Diferentemente da maioria dos kernels, o Linux leva em consideração os bits TOS (Type-Of-Service) nos pacotes IP e dá prioridade aos pacotes que estão rotulados como interativos (baixa latência). Agora a enrascada! Infelizmente, problemas mentais por parte da Microsoft fazem com que tenhamos de desativar esse comportamento perfeitamente razoável.

Todos os pacotes originários do Windows 95, 98, NT e 2000 são rotulados como interativos, independentemente de seus propósitos. Por outro lado, os sistemas UNIX, normalmente não marcam nenhum pacote como interativo. Se seu gateway Linux atende a uma rede mista de sistemas UNIX e Windows, os pacotes do Windows receberão incondicionalmente tratamento preferencial. O impacto no desempenho para sistemas UNIX pode ser bem perceptível.

Podemos desativar a classificação de pacotes baseados em TOS ao compilarmos o kernel Linux. Basta dizermos não à opção "IP: use TOS value as routing key".

Quando o mascaramento IP (NAT) estiver habilitado, ele informa o kernel para remontar fragmentos de pacotes num pacote completo antes de encaminhá-los, mesmo que o kernel tenha de fragmentar o pacote novamente, de imediato, para poder enviá-lo. Essa remontagem pode significar um número razoável de ciclos de CPU, mas hoje em dia as CPUs são suficientemente rápidas, de modo que isto realmente não se constitui em problema para as máquinas modernas.

O Linux permite que alteremos os endereços em nível MAC de certos tipos de interfaces de rede:

```
redhat% ifconfig eth1
eth1  Link encap:Ethernet HWaddr 00:02:B3:19:C8:87
      BROADCAST MULTICAST MTU:1500 Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      Interrupt:7 Base address:0xee80
redhat% sudo ifconfig eth1 hw ether 00:02:B3:19:C8:21
redhat% ifconfig eth1
eth1  Link encap:Ethernet HWaddr 00:02:B3:19:C8:21
      BROADCAST MULTICAST MTU:1500 Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
```

Snow666


```
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
Interrupt:7 Base address:0xee80
```

Trata-se de um recurso perigoso que tende a danificar as coisas. Ele pode ser prático, porém deve ser usado somente como um último recurso.

13.15 Leitura Recomendada

STEVENS, W. RICHARD. *TCP/IP Illustrated, Volume One: The Protocols.* Reading, MA: Addison-Wesley. 1994.
WRIGHT, GARY R.; W. RICHARD STEVENS. *TCP/IP Illustrated, Volume Two: The Implementation.* Reading, MA: Addison-Wesley. 1995.
Esses dois livros são guias completos e excelentes sobre a camada de protocolo TCP/IP. Um pouco defasados, porém ainda consistentes.
STEVENS, W. RICHARD. *UNIX Network Programming.* Upper Saddle River, NJ: Prentice Hall. 1990.
STEVENS, W. RICHARD. *UNIX Network Programming, Volume 1: Networking APIs—Sockets and XTI.* Upper Saddle River, NJ: Prentice Hall. 1997.
STEVENS, W. RICHARD. *UNIX Network Programming, Volume 2: Interprocess Communications.* Upper Saddle River, NJ: Prentice Hall. 1998.

Esses livros são as bíblias dos estudantes em cursos de rede que envolvam programação. Caso precise somente da interface de soquetes Berkeley, a edição original é uma excelente referência. Caso precise também da interface STREAMS, então a segunda edição, que se tornou muito grande para ser colocado num só volume, é uma boa opção. A segunda edição também inclui IPv6. Todos os três são escritos de maneira clara num estilo típico de Rich Stevens.

TANENBAUM, ANDREW. *Computer Networks. 3. ed.* Upper Saddle River, NJ: Prentice Hall. 1996.

Esse foi o primeiro texto sobre redes e ainda é um clássico. Ele contém uma descrição bem completa de todos os detalhes internos que se passam nas camadas física e de link da pilha de protocolos. Edições iniciais eram um tanto "amarradas" aos protocolos ISO, porém a última edição foi atualizada para abranger a Internet moderna.

SALUS, PETER H. *Casting the Net, From ARPANET to INTERNET and Beyond.* Reading, MA: Addison-Wesley. 1995.

Essa é uma história apaixonante da ARPANET que conta como ela foi se transformando na Internet, escrita por um historiador que tem estado com o pessoal do UNIX tempo suficiente para se confundir com um deles!

COMER, DOUGLAS. *Internetworking with TCP/IP Volume 1: Principles, Protocols, and Architectures. 4. ed.. Upper Saddle River, NJ: Prentice Hall. 2000.*

Os livros de Comer foram por um longo tempo a referência-padrão para protocolos TCP/IP.
Essa nova edição apresenta descrições de modernas tecnologias de rede, bem como a família de protocolos TCP/IP. Ele é feito na forma de um livro didático e é uma excelente fonte de material para embasamento.

HEDRICK, CHARLES. "Introduction to the Internet Protocols." Rutgers University.

Esse documento é uma introdução suave ao TCP/IP. Ele não parece ter um endereço permanente, porém é distribuído amplamente na Web; procure-o.

HUNT, CRAIG. *TCP/IP Network Administration. 2. ed.* Sebastopol, CA: O'Reilly & Associates. 1998.

Como outros livros do tipo lacônico, este se destina a administradores de sistemas UNIX. Metade do livro é sobre TCP/IP e o restante trata de tópicos do UNIX de nível mais alto como e-mail e login remoto.
www.netscan.org mantém uma lista de amplificadores de ataques "smurf" (sistemas que respondem a ICMP de difusão ampla). Você poderá digitar o endereço IP de sua rede e verificar se ela está OK, desde que esteja utilizando sub-redes nos limites de bytes. Se sua rede se encontrar nessa lista, desabilite a difusão ampla direcionada conforme descrito neste capítulo para se tornar um cidadão melhor da Net.
Uma excelente coleção de documentos sobre a historia da Internet e suas várias tecnologias pode ser encontrada em www.isoc.org/internet/history.

Snow666

13.16 Exercícios

E13.1 Como o atendimento a (isto é, a obediência a) redirecionamentos ICMP permite que um usuário não autorizado comprometa a rede?

E13.2 O que é MTU de um link de rede? O que acontece se o MTU para um dado link for configurado com um valor muito alto? E muito baixo?

★ **E13.3** Após instalar um sistema Linux novinho em folha, como você poderia solucionar as questões relativas à segurança mencionadas neste capítulo? Verifique se algum destes problemas de segurança foram solucionados nos seus sistemas Linux. (Requer acesso root.)

★ **E13.4** Explique o conceito de sub-redes e explique a razão de sua utilidade. O que são máscaras de rede? Como as máscaras de rede se relacionam com a divisão das partes rede e host de um endereço IP?

★ **E13.5** A rede 134.122.0.0/16 foi subdividida em /19 redes.

a) Quantas redes /19 existem? Enumere-as. Quais são suas máscaras de rede?
b) Quantos hosts poderia haver em cada rede?
c) Determine à qual rede o endereço IP 134.122.67.124 pertence.
d) Qual é o endereço de difusão ampla para cada rede?

★ **E13.6** O host 128.138.2.4 na rede 128.138.2.0/24 deseja enviar um pacote para o host 128.138.129.12 na rede 128.138.129.0/24. Suponha o seguinte:

- O host 128.138.2.4 possui uma rota-padrão através de 128.138.2.1.
- O host 128.138.2.4 acaba de ser inicializado e não enviou nem recebeu nenhum pacote.
- Todas as outras máquinas na rede já estão rodando por um longo período de tempo.
- O roteador 128.138.2.1 possui um link direto com 128.138.129.1, o gateway para a sub-rede 128.138.129.0/24.

a) Liste todas as etapas necessárias para enviar o pacote. Mostre os endereços Ethernet e IP de origem e de destino de todos os pacotes transmitidos.
b) Se a rede fosse 128.138.0.0/16, a sua resposta mudaria? Como?
c) Se a rede 128.138.2.0 fosse uma rede /26 em vez de uma /24, a sua resposta mudaria? Em que sentido?

★★ **E13.7** Quais são as etapas necessárias para acrescentar-se uma máquina nova à rede de seu ambiente? Na resposta, use parâmetros apropriados para a sua rede e situação local. Suponha que a nova máquina já rode Linux.

★★ **E13.8** Mostre o arquivo de configuração necessário para configurar um servidor DHCP que atribui endereços no intervalo 128.138.192.[1-55]. Use um tempo de arrendamento de duas horas e garanta que o host com endereço Ethernet 00:10:5A:C7:4B:89 sempre receba o endereço IP 128.138.192.55.

www.snow666.blogspot.com
Snow666

14 Roteamento

Controlar para onde o tráfego de rede deve fluir não é uma tarefa fácil. O Capítulo 13 apresentou uma pequena introdução ao encaminhamento de pacotes IP. Neste capítulo, examinamos o processo de encaminhamento com mais detalhes e investigamos vários protocolos de rede que permitem que os roteadores descubram automaticamente rotas eficientes. Os protocolos de roteamento não somente reduzem o encargo administrativo do dia-a-dia de manter informações de roteamento, como também permitem que o tráfego de rede seja redirecionado rapidamente caso um roteador ou rede falhe.

É importante fazer a distinção entre o processo de efetivamente encaminhar pacotes IP e o gerenciamento da tabela de roteamento que dirige este processo, ambos comumente denominados de "roteamento". O encaminhamento de pacotes é simples, ao passo que o cômputo da rota é cheio de truques; conseqüentemente, o segundo significado é usado com mais freqüência na prática. Este capítulo descreve somente o roteamento unidifusão; o roteamento multidifusão envolve uma série de problemas muito diversos e está além do escopo deste livro.

Convencionalmente diz-se que o roteamento IP é excepcionalmente difícil, compreendido apenas por alguns poucos hippies cabeludos que vivem nos dutos do sistema de aquecimento sob o campus dos Laboratórios Lawrence Berkeley na Califórnia. Na realidade, este não é o caso, desde que você entenda a premissa básica de que o roteamento IP é o roteamento do "próximo salto". A qualquer dado momento, você somente precisa determinar o *próximo* host ou roteador na jornada de um pacote até o seu destino final. Esta é uma abordagem diferente daquela de muitos protocolos mais antigos que determinam o caminho exato que um pacote percorrerá antes de deixar seu host original, um esquema conhecido como roteamento de origem.[1]

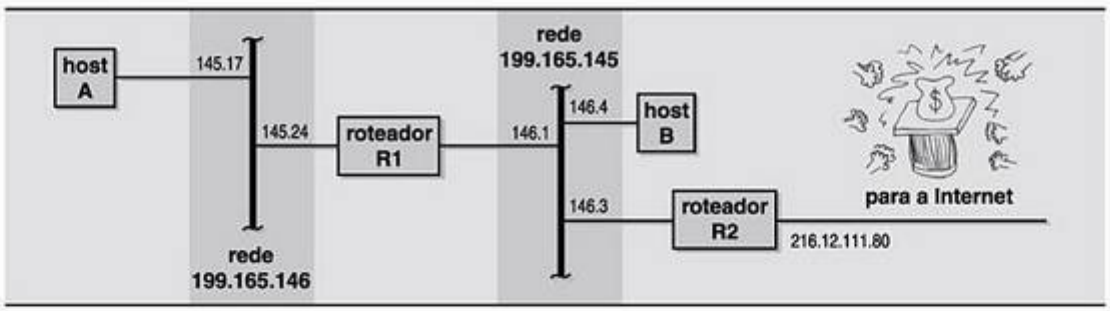
14.1 Encaminhamento de Pacotes: Uma Visão Mais Detalhada

Antes de pularmos para o gerenciamento de tabelas de roteamento, vamos dar uma olhada com mais detalhes na maneira pela qual as tabelas são usadas. Consideremos a rede ilustrada no Diagrama A.

1. Também é possível a pacotes IP de roteamento de origem, porém isto quase nunca é feito. O recurso não é amplamente aplicado devido a questões de segurança.

Snow666

Diagrama A Exemplo de redes.



O roteador R1 conecta duas redes Ethernet e o roteador R2 conecta uma das redes ao mundo externo. (Por enquanto, assumiremos que R1 e R2 são computadores com Linux, e não roteadores dedicados.) Vamos dar uma olhada em algumas tabelas de roteamento e algumas situações específicas de encaminhamento de pacotes. Primeiramente, apresentamos a tabela de roteamento do host A:

```
A% netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags  MSS  Window  irtt  Iface
199.165.145.0   0.0.0.0         255.255.255.0   U        0  0          0  eth0
127.0.0.0       0.0.0.0         255.0.0.0       U        0  0          0  lo
0.0.0.0         199.165.145.24  0.0.0.0         UG       0  0          0  eth0
```

*Verifique o Capítulo 13 para obter maiores informações sobre **ifconfig**.*

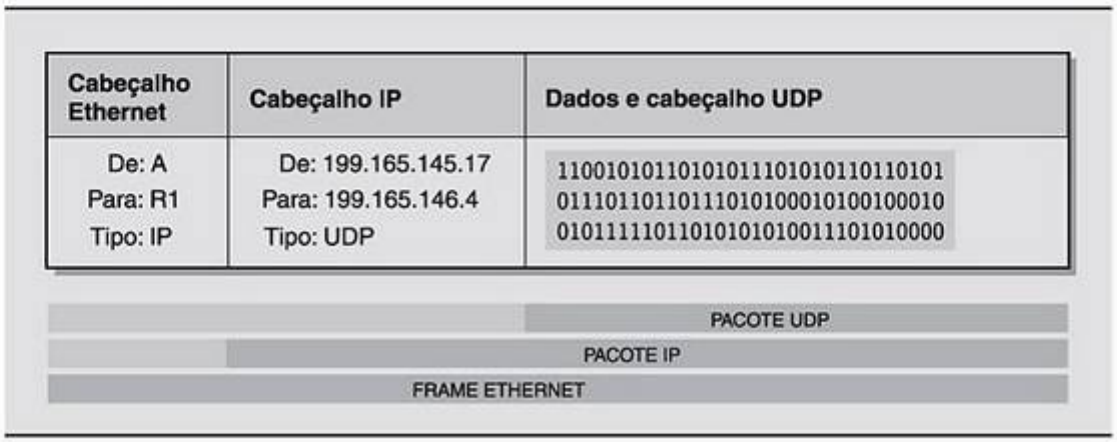
O host A possui a configuração de roteamento mais simples das quatro máquinas. As duas primeiras rotas descrevem as interfaces de rede da própria máquina em termos de roteamento-padrão. Essas entradas existem para que o encaminhamento a redes conectadas diretamente não precise ser tratado como um caso especial. eth0 é a interface Ethernet do host A e lo é a interface de loopback, uma interface de rede virtual emulada por software. Entradas como essas são normalmente adicionadas de forma automática por **ifconfig** quando a interface de rede é configurada.

A rota-padrão no host A encaminha todos os pacotes não destinados ao endereço de loopback ou para a rede 199.165.145 para o roteador R1, cujo endereço nesta rede é 199.165.145.24. O flag G indica que essa rota vai para um gateway, e não para uma das interfaces locais de A. Os gateways podem estar somente a um salto de distância.

Veja o Capítulo 13 para obter maiores informações sobre endereçamento.

Suponha que um processo em A envie um pacote para B, cujo endereço é 199.165.146.4. A implementação IP procura uma rota para a rede de destino, 199.165.146, mas nenhuma delas atende às especificações. A rota-padrão é chamada e o pacote é encaminhado para R1. O Diagrama B ilustra o pacote que sai efetivamente da Ethernet (os endereços contidos no cabeçalho Ethernet são os endereços MAC das interfaces de A e de R1 na rede 145).

Diagrama B Pacote Ethernet



O endereço de hardware de destino Ethernet é aquele do roteador R1, porém o pacote IP oculto dentro do frame Ethernet não menciona R1 em si. Quando R1 inspeciona o pacote que ele recebeu, verá pelo endereço IP de

Snow666

destino que não se trata do último destino do pacote. Ele usará então sua própria tabela de roteamento para encaminhar o pacote para o host B sem regravar o cabeçalho IP, de forma que ele ainda mostra o pacote proveniente de A.

Eis a tabela de roteamento para o host R1:

```
R1% netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags MSS   Window   irtt  Iface
127.0.0.0       0.0.0.0         255.0.0.0       U       0   0           0  lo
199.165.145.0   0.0.0.0         255.255.255.0   U       0   0           0  eth0
199.165.146.0   0.0.0.0         255.255.255.0   U       0   0           0  eth1
0.0.0.0         199.165.146.3   0.0.0.0         UG      0   0           0  eth1
```

Essa tabela é similar àquela do host A, exceto pelo fato de que há duas interfaces de rede físicas. A rota-padrão nesse caso aponta para R2, já que é o gateway através do qual a Internet pode ser atingida. Pacotes associados a qualquer uma das redes 199.165 podem ser encaminhados diretamente.

Da mesma maneira que o host A, o host B possui apenas uma interface de rede real. Entretanto, B precisa de uma rota adicional para funcionar corretamente pois ela possui conexões diretas para dois roteadores diferentes. O tráfego para a rede 199.165.145 tem de passar por R1, ao passo que outros dados têm de sair para a Internet através de R2.

```
B% netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags MSS   Window   irtt  Iface
127.0.0.0       0.0.0.0         255.0.0.0       U       0   0           0  lo
199.165.145.0   199.165.146.1   255.255.255.0   U       0   0           0  eth0
199.165.146.0   0.0.0.0         255.255.255.0   U       0   0           0  eth0
0.0.0.0         199.165.146.3   0.0.0.0         UG      0   0           0  eth0
```

Verifique o Capítulo 13 para obter uma explicação sobre redirecionamentos ICMP.

É possível configurar o host B com conhecimento inicial de somente um gateway e, portanto, dependente da ajuda de redirecionamentos ICMP para eliminar saltos extras. Por exemplo, uma configuração inicial para o host B é:

```
B% netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags MSS   Window   irtt  Iface
127.0.0.0       0.0.0.0         255.0.0.0       U       0   0           0  lo
199.165.146.0   0.0.0.0         255.255.255.0   U       0   0           0  eth0
0.0.0.0         199.165.146.3   0.0.0.0         UG      0   0           0  eth0
```

Se B enviar então um pacote para o host A (199.165.145.17), nenhuma rota será concordante e o pacote será encaminhado para R2 para entrega. R2 (que, por ser um roteador, provavelmente tem informações completas sobre a rede) enviará o pacote para R1. Já que R1 e B se encontram na mesma rede, R2 também enviará um aviso de redirecionamento ICMP para B e B introduzirá uma rota host para A em sua tabela de roteamento:

```
199.165.145.17  199.165.146.1   255.255.255.255 UGHD     0  0           0  eth0
```

Essa rota enviará todo o tráfego futuro para A diretamente através de R1. Entretanto, ela não afeta o roteamento para os demais hosts na rede de A, todos os quais terão de ser roteados por redirecionamentos distintos a partir de R2.

Algumas instalações optaram pelos redirecionamentos ICMP como "protocolo" de roteamento principal, imaginando que esta forma seria muito dinâmica. Infelizmente, assim que o kernel toma conhecimento de uma rota por meio de um redirecionamento, a rota tem de ser deletada manualmente, ou a máquina tem de ser reinicializada caso estas informações mudem. Devido a esse problema e a várias outras desvantagens de redirecionamentos (aumento de carga na rede, carga aumentada em R2, desorganização completa na tabela de roteamento, dependência de servidores extras), não recomendamos o emprego de redirecionamentos para configurações como essas. Numa rede configurada apropriadamente, os redirecionamentos jamais devem aparecer na tabela de roteamento.

14.2 Daemons de Roteamento e Protocolos de Roteamento

Em redes simples como aquela ilustrada no Diagrama A, é perfeitamente razoável configurar o roteamento manualmente. Entretanto, em algum momento, as redes se tornam muito complexas para serem gerenciadas dessa maneira

www.snow666.blogspot.com
Snow666

(possivelmente devido à sua taxa de crescimento). Em vez de ter de informar explicitamente cada computador em cada uma das redes como atingir cada um dos demais computadores e redes, seria interessante se os computadores pudessem simplesmente unir suas inteligências e descobrir tudo isto. Essa é a tarefa dos protocolos de roteamento e dos daemons que os implementam.

Os protocolos de roteamento possuem uma grande vantagem em relação a sistemas de roteamento estáticos já que eles reagem e se adaptam a mudanças das condições da rede. Se um link cair, os daemons de roteamento poderão descobrir e propagar rapidamente rotas alternativas para as redes que o link servia, caso realmente existisse um.

Os daemons de roteamento coletam informações de três fontes: arquivos de configuração, tabelas de roteamento existentes e daemons de roteamento em outros sistemas. Essas informações são mescladas para computar um conjunto de rotas otimizado, e as novas rotas são então realimentadas na tabela de roteamento do sistema (e possivelmente alimentadas para outros sistemas através de um protocolo de roteamento). Pelo fato de as condições de rede mudarem ao longo do tempo, os daemons de roteamento devem verificar periodicamente entre si para assegurarem-se de que suas informações de roteamento ainda estão válidas.

A maneira exata pela qual as rotas são computadas depende do protocolo de roteamento. Hoje em dia são dois os protocolos comumente em uso: protocolos de vetor de distância e protocolos de estado de link.

Protocolos de vetor de distância

Os protocolos de vetor de distância (também conhecidos como "fofoqueiros") se baseiam na idéia geral "Se o roteador X se encontra a cinco saltos de distância da rede Y e eu me encontro numa posição adjacente ao roteador X, então tenho de estar a seis saltos da rede Y". Você anuncia a que distância se encontra das redes sobre as quais tem conhecimento. Se seus vizinhos não tiverem conhecimento de uma maneira melhor de atingir cada uma das redes, eles marcam você como o melhor gateway. Se eles já conhecerem uma rota mais curta, eles ignoram seus anúncios[2]. Com o passar do tempo, as tabelas de roteamento supostamente convergem para um estado estável.

Trata-se realmente de uma idéia muito refinada. Se ela funcionasse da forma divulgada, o roteamento seria relativamente simples. Infelizmente esse tipo de algoritmo não é capaz de lidar bem com mudanças na topologia. Em alguns casos, loops infinitos (por exemplo, o roteador X recebe informações do roteador Y e as envia pra o roteador Z que, por sua vez, as envia de volta para o roteador Y) podem impedir que as rotas convirjam de fato. Os protocolos de vetor de distância do mundo prático têm de evitar tais problemas introduzindo heurística complexa ou forçando restrições arbitrárias como a noção do RIP (Routing Information Protocol, protocolo de informações de roteamento) em que toda rede com mais de 15 saltos de distância é inatingível.

Mesmo em casos não patológicos, pode levar vários ciclos de atualização para que todos os roteadores atinjam um estado estável. Portanto, para garantir que o roteamento não ficará empacado por um longo período, o tempo de ciclo tem de ser curto, e por esta razão os protocolos de vetor de distância como um todo tendem a ser prolixos. Por exemplo, o RIP requer que os roteadores façam uma difusão ampla de todas as suas informações de roteamento a cada 30 segundos. O IGRP e o EIGRP enviam atualizações a cada 90 segundos.

Por outro lado, o BGP (Border Gateway Protocol, protocolo de gateway de fronteira) transmite toda a tabela de uma só vez e depois transmite as modificações à medida que elas forem ocorrendo. Essa otimização reduz substancialmente o potencial para tráfego "de bate-papo" (e, em sua maioria, desnecessário).

A Tabela 14.1 lista os protocolos de vetor de distância que são usados comumente hoje em dia.

Protocolos de estado de link

Os protocolos de estado de link distribuem informações de uma maneira relativamente não processada. Os registros negociados entre os roteadores são normalmente do tipo "O roteador X é adjacente ao roteador Y e o

Tabela 14.1 Protocolos de roteamento de vetor de distância comuns.

Protocolo	Nome por extenso	Aplicação
RIP	Routing Information Protocol	LANs internas
IGRP	Interior Gateway Routing Protocol	WANs pequenas
EIGRP	Enhanced Interior Gateway Routing Protocol	WANs, LANs corporativas
BGP	Border Gateway Protocol	Roteamento de backbone Internet

2. Na verdade, não é tão simples assim, já que há regras para manipulação de alterações na topologia que podem aumentar a distância de rotas existentes. Alguns protocolos de vetor de distância como o EIGRP mantêm informações sobre várias rotas possíveis de modo que elas sempre têm um plano de emergência. Os detalhes exatos não são importantes.

Snow666

link está ativo". Um conjunto completo de tais registros forma um mapa de conectividade da rede por meio do qual cada roteador pode computar sua própria tabela de roteamento. A principal vantagem dos protocolos de estado de link em relação aos protocolos de vetor de distância é a capacidade de convergir rapidamente para uma solução de roteamento operacional após a ocorrência de alguma pane. A desvantagem é que manter um "mapa" completo da rede em cada nó requer memória e poder de processamento que não seriam necessários num sistema de roteamento de vetor de distância.

Pelo fato de as comunicações entre os roteadores em um protocolo de estado de link não fazerem parte do algoritmo de cômputo das rotas em si, elas podem ser implementadas de tal maneira que os loops de transmissão não ocorram. As atualizações para o banco de dados de topologia se propagam através da rede de modo eficiente, a um custo menor em termos de largura de banda da rede e de tempo de CPU.

Os protocolos de estado de link tendem a ser mais complicados do que os protocolos de vetor de distância, porém isto pode ser explicado em parte pelo fato de os protocolos de estado de link facilitarem a implementação de recursos avançados como o roteamento de tipos de serviço e rotas múltiplas para o mesmo destino. Nenhum desses recursos é suportado em sistemas Linux comuns; você tem de usar roteadores dedicados para tirar proveito deles.

Os protocolos de estado de link comuns são mostrados na Tabela 14.2.

Tabela 14.2 Protocolos de roteamento de estado de link comuns.

Protocolo	Nome por extenso	Aplicação
OSPF	Open Shortest Path First	LANs internas, WANs pequenas
IS-IS	Intermediate System to Intermediate System	ISPs, Grandes Redes IP

Métrica de custo

Para que um protocolo de roteamento determine qual caminho para uma dada rede é o mais curto, ele tem de definir o que é conhecido como o "mais curto".[3] É o caminho envolvendo o menor número de saltos? O caminho de menor latência? A maior largura de banda intermediária mínima? O de custo financeiro mais baixo?

Para fins de roteamento, a qualidade de um link é representada por um número chamado de métrica de custo. Juntando os custos de cada link num caminho, o custo de um caminho poderia ser calculado. Nos sistemas mais simples, cada link tem um custo unitário, levando a contagem de saltos como uma métrica de caminho. Porém qualquer uma das considerações mencionadas pode ser convertida numa métrica de custo numérica.

Os "experts" em redes trabalharam arduamente e por longo tempo para tornar a definição de métrica de custo flexível, e alguns protocolos modernos até mesmo permitem que sejam usadas diferentes métricas para diferentes tipos de tráfego de rede. Não obstante, em 99% dos casos, todo esse árduo trabalho pode ser ignorado com segurança. A métrica-padrão funciona perfeitamente bem para a maioria dos sistemas. Você poderá se deparar com situações nas quais o verdadeiro caminho mais curto para um determinado destino possa não ser uma boa rota-padrão por questões políticas. Para tratar desses casos, você poderá aumentar artificialmente o custo de links críticos para fazer com que eles pareçam menos atraentes. Não mexa no restante da configuração de roteamento.

Protocolos internos e externos

Um "sistema autônomo" é um grupo de redes sob o controle político-administrativo de uma única entidade. A definição é vaga; sistemas autônomos do mundo real podem ser grandes como a rede corporativa mundial ou pequenos como uma rede num prédio ou de um único departamento acadêmico. Tudo depende de como você quer gerenciar o roteamento. A tendência genérica é a de fazer com que os sistemas autônomos tenham o maior tamanho possível. Essa convenção simplifica a administração e torna o roteamento o mais eficiente possível.

O roteamento no interior de um sistema autônomo é um tanto diverso do roteamento entre sistemas autônomos. Os protocolos para roteamento entre sistemas autônomos (protocolos "externos") normalmente têm de lidar com rotas de várias redes e eles têm de lidar de forma graciosa com o fato de que os roteadores vizinhos estão sob o controle de uma outra pessoa. Os protocolos externos não revelam a topologia dentro de um sistema autônomo, no sentido de que eles podem ser imaginados como um segundo nível de hierarquia de roteamento que lida com conjuntos de redes em vez de cabos ou hosts individuais.

3. Felizmente, ele não tem de definir qual é o significado de "é".

Snow666

Na prática, instalações pequenas à média raramente precisam executar um protocolo externo a menos que eles estejam conectados a mais do que um ISP. Com vários ISPs, a fácil divisão de redes em domínios locais e Internet se reduz e os roteadores têm de decidir qual rota para a Internet é a melhor para um dado endereço. (Entretanto, isso não quer dizer que *cada* um dos roteadores tenha de conhecer essas informações. A maioria dos hosts pode permanecer ignorante e rotear os seus pacotes (rota-padrão) através de um gateway que esteja melhor informado.)

Embora os protocolos externos não sejam tão diferentes de seus equivalentes internos, este capítulo se concentra nos protocolos internos e nos daemons que os suportam. Se sua instalação também tiver de usar um protocolo externo, veja a lista de literatura recomendada no final deste capítulo para obter algumas referências sugeridas.

14.3 Protocolos na Parada

Vários protocolos de roteamento internos estão sendo usados hoje em dia. Nesta seção, introduziremos os principais concorrentes e sintetizaremos seus principais pontos fortes e fracos.

RIP (Routing Information Protocol)

O RIP, definido na RFC1058, é um antigo protocolo da Xerox que foi adaptado para redes IP.

Ele é o protocolo usado por **routed**. O RIP é um protocolo de vetor de distância simples que usa contagem de saltos como métrica de custo. Pelo fato de ter sido desenvolvido numa era em que um único computador custava centenas de milhares de dólares e as redes eram relativamente pequenas, o RIP considera qualquer host a 15 ou mais saltos de distância como inatingível. Portanto, grandes redes locais que possuam mais do que 15 roteadores ao longo de qualquer trajeto não poderão usar o protocolo RIP.

Embora o RIP seja um glutão em termos de consumo de recursos devido ao seu uso abusivo de difusão ampla, ele faz um excelente trabalho quando uma rede é modificada com freqüência ou quando a topologia de redes remotas não é conhecida. Porém, pode ser lento para estabilizar após um link cair.

O RIP é implementado amplamente em plataformas outras que não o Linux. Uma série de dispositivos comuns que vão desde impressoras a componentes de rede gerenciados por SNMP são capazes de ouvir anúncios RIP para tomar conhecimento de possíveis gateways. Além disso, **routed** está disponível para todas as versões do Linux, de modo que o RIP é de fato o menor denominador comum em termos de protocolo de roteamento. Muitas vezes o RIP é usado para roteamento de LANs, e um protocolo com mais recursos é utilizado para redes de longa distância.

Consulte o Capítulo 13 para obter maiores informações sobre endereçamento sem o emprego de classes, também conhecido como CIDR.

RIP-2: Routing Information Protocol, versão 2

O RIP-2 é uma revisão moderada do RIP que acrescenta suporte a alguns poucos recursos que faltavam no protocolo original. A modificação mais importante é que o RIP-2 distribui máscaras de rede juntamente com os endereços de próximo salto, de modo que ele fornece melhor suporte para redes divididas em subredes e CIDR. Um vago aumento na segurança do RIP também foi incluído, porém a definição de um sistema de autenticação específico foi postergada para o futuro.

Muitas instalações usam **routed** em seu modo **-q** ("quiet"), no qual ele gerencia a tabela de roteamento e escuta atualizações de roteamento na rede, mas não difunde de maneira ampla qualquer informação própria. Nessas instalações os cálculos de rotas em si são normalmente executados com um protocolo mais eficiente como o OSPF (veja a seguir). As rotas computadas são convertidas em atualizações RIP para consumo por parte de máquinas não roteadoras. **routed** é leve (no modo **-q**) e suportado universalmente, de forma que a maior parte das máquinas pode desfrutar dos benefícios do roteamento dinâmico sem qualquer configuração especial.

O RIP-2 fornece vários recursos que parecem destinados a esse ambiente multiprotocolo. As atualizações "Próximo salto" permitem aos difusores anunciarem rotas para as quais eles não são o gateway real, e "tags de rotas" permitem que rotas descobertas externamente sejam propagadas através do RIP.

O RIP-2 pode ser executado num modo de compatibilidade que preserva a maioria dos novos recursos de RIP-2 sem abandonar inteiramente os receptores RIP comuns. Na maioria dos aspectos, o RIP-2 é idêntico ao RIP e deve ter preferência em relação ao RIP se for suportado pelos sistemas que você estiver usando. Entretanto, as distribuições Linux geralmente não oferecem suporte a ele na configuração que vem de fábrica.

OSPF: Open Shortest Path First

O OSPF é definido na RFC2328. Trata-se de um protocolo de estado de link. "Shortest path first", ou seja, o caminho mais curto primeiro, se refere ao algoritmo matemático usado para calcular as rotas; "open", ou seja, aberto, é usado no sentido de "não proprietário".

www.snow666.blogspot.com
Snow666

O OSPF foi o primeiro protocolo de roteamento de estado de link a ser amplamente usado e ainda é o mais popular. Sua adoção em larga escala foi impelida em grande parte pelo seu suporte no **gated**, um daemon de roteamento multiprotocolo popular sobre o qual falaremos a respeito mais adiante. Infelizmente, o protocolo em si é muito complexo e, portanto, só vale a pena em instalações de tamanho significativo (em que o comportamento do protocolo de roteamento realmente faz diferença).

A especificação do protocolo OSPF não impõe nenhuma métrica de custo em particular. A implementação de **gated** usa, como padrão, contagens de saltos, assim como a implementação da Cisco. Os roteadores Cisco também podem ser configurados para usarem largura de banda de rede como uma métrica de custo.

O OSPF é um protocolo de peso industrial que funciona bem para topologias grandes e complicadas. Ele oferece várias vantagens em relação ao RIP, inclusive a capacidade de controlar vários caminhos para um único destino e a habilidade de dividir a rede em seções ("áreas") que compartilham somente informações de roteamento de alto nível.

IGRP e EIGRP: Interior Gateway Routing Protocol

O IGRP e seu sucessor "envenenado", o EIGRP, são protocolos de roteamento proprietários que rodam somente em roteadores Cisco. O IGRP foi criado para atender a algumas das deficiências do RIP antes da existência de padrões robustos como o OSPF. O EIGRP é configurado de forma semelhante ao IGRP, embora ele seja, na verdade, bem diferente em seu projeto de protocolo subjacente. O IGRP trata somente anúncios de roteamento usando fronteiras de classes de endereços IP tradicionais, ao passo que o EIGRP entende máscaras de rede CIDR arbitrárias.

Tanto o IGRP quanto o EIGRP são protocolos de vetor de distância, porém eles são projetados para evitar os problemas de looping e de convergência encontrados em outros sistemas de vetor de distância. O EIGRP em particular é amplamente considerado um modelo de perfeição do roteamento de vetor de distância. Para a maioria dos propósitos, EIGRP e OSPF são igualmente funcionais.

Em nossa opinião, é melhor aderir a um protocolo de roteamento estabelecido, não proprietário e bem proliferado como o OSPF. Há um número maior de pessoas usando e trabalhando no OSPF do que no EIGRP e há várias implementações disponíveis.

IS-IS: o "padrão" ISO

IS-IS (Intra-domain Intermediate System to Intermediate System Routeing Protocol) é a resposta da International Organization for Standardization ao OSPF. Ele foi projetado originalmente para gerenciar "roteamento" para protocolos de rede OSI e foi posteriormente estendido para manipular roteamento IP.

Tanto o IS-IS quanto o OSPF foram desenvolvidos no início dos anos 90 na época em que os protocolos ISO estavam politicamente em voga. Uma atenção logo no início por parte da IETF ajudou a atribuir ao IS-IS um aspecto de legitimidade para IP, mas ele parece se distanciar cada vez mais do OSPF em termos de popularidade.

Hoje, o emprego do IS-IS é raro. O protocolo em si é emaranhado com grande quantidade de "bagagem" ISO e, geralmente, deve ser evitado.

MOSPF, DVMRP e PIM: protocolos de roteamento multidifusão

MOSPF (Multicast OSPF, OSPF multidifusão), DVMRP (Distance Vector Multicast Routing Protocol, protocolo de roteamento multidifusão de vetor de distância) e PIM (Protocol Independent Multicast, multidifusão independente de protocolo) são protocolos desenhados para oferecer suporte à multidifusão IP, uma tecnologia que ainda não é amplamente empregada. Você poderá encontrar indicadores para informações adicionais sobre esses protocolos em www.mbone.com.

RDP

O RDP (Router Discovery Protocol, protocolo de descoberta de roteadores) usa mensagens ICMP enviadas ao endereço IP multidifusão 224.0.0.1 para anunciar e tomar conhecimento sobre outros roteadores numa rede. Infelizmente, nem todos os roteadores fazem esses anúncios atualmente e nem todos os hosts reconhecem esses anúncios. A esperança é que algum dia este protocolo se torne mais popular.

14.4 ROUTED: ABRA NOVOS CAMINHOS POR CONTA PRÓPRIA

Talvez você não seja rico. Pode ser que você não tenha uma boa aparência. Mas certamente você sempre usou **routed**. **routed** foi por um bom tempo um daemon-padrão de roteamento no UNIX e ainda é fornecido com a maioria das versões do UNIX e no Linux.

Snow666

O comando **routed** do Linux fala somente RIP. Caso precise de RIP-2, sempre é possível usar **gated** em seu lugar como um daemon RIP-2. Ele somente é um pouco mais complicado, especialmente se você tiver que instalá-lo por conta própria. (Você realmente só precisará do RIP-2 caso tenha subredes com comprimentos de máscaras diferentes.)

routed pode ser executado no modo servidor (**-s**) ou no modo silencioso (**-q**). Ambos os modos escutam difusões amplas, porém somente servidores distribuem suas próprias informações. Em geral, somente máquinas com várias interfaces devem ser servidores. Se não for especificado **-s** nem **-q**, assume-se que **routed** deva ser executado no modo silencioso com uma interface e no modo servidor com mais de uma interface. Porém em muitos sistemas, esse recurso apresenta problemas. [4]

*Veja o Capítulo 13 para obter maiores informações sobre **route**.*

routed acrescenta as rotas por ele descobertas à tabela de roteamento do kernel. As rotas têm de ser ouvidas novamente pelo menos a cada quatro minutos, caso contrário serão removidas. Entretanto, **routed** sabe quais rotas ele acrescentou e não eliminará rotas estáticas que foram instaladas por meio do comando **route**.

routed -t pode ser usado para depurar o roteamento. Essa opção faz com que **routed** seja executado em primeiro plano e imprima todos os pacotes que ele envia ou recebe.

routed normalmente descobre dinamicamente informações de roteamento e não requer configuração. Porém, se a sua instalação contiver gateways para a Internet ou para outros sistemas autônomos, pode ser que você tenha de acrescentar alguns passos adicionais para fazer com que estes links funcionem com **routed**.

Se você tiver somente um único gateway de fronteira, poderá anunciá-lo como uma rota-padrão global executando seu comando **routed** com o flag **-g**. Isso equivale a configurar a rota-padrão numa única máquina, exceto pelo fato de ela ser propagada através de sua rede.

routed também suporta um arquivo de configuração, **/etc/gateways**, que foi desenvolvido para fornecer informações estáticas sobre gateways para "pré-carga" na tabela de roteamento à qual foi aplicado **routed**. Nos tempos atuais, caso precise dessa funcionalidade, você realmente deve executar o comando **gated** em seu lugar.

gated pode ser obtido em www.gated.org.

14.5 GATED: UM DAEMON DE ROTEAMENTO MELHOR

gated é uma estrutura de roteamento genérica que permite que diversos protocolos de roteamento diferentes sejam usados simultaneamente. **gated** fornece controle de grande precisão sobre rotas anunciadas, endereços de difusão ampla, políticas de confiabilidade e métrica. Ele pode compartilhar rotas entre diversos protocolos, permitindo que os gateways de roteamento sejam construídos entre áreas que foram padronizadas em sistemas de roteamento diferentes. **gated** também possui um dos projetos mais interessantes em termos de interfaces para administração e de arquivo de configuração entre os programas administrativos Linux.

Muitas pessoas contribuíram para o desenvolvimento de **gated**. O trabalho foi originalmente coordenado pela Cornell University. **gated** começou como um software para distribuição gratuita, porém em 1992 ele foi privatizado e adquirido pelo Merit GateD Consortium. As versões atuais de **gated** estão disponíveis somente para membros do consórcio. A associação como membro (dez categorias! quatro linhas de produtos! mas espere! você também tem direito a ...) é aberta a todo o mundo, porém ela requer a efetivação de um acordo de licenciamento e é cara para usuários não acadêmicos.

Embora a definição de "uso acadêmico e para pesquisa" seja relativamente ampla, recomendamos ficar longe do "rolo" no qual **gated** acabou se envolvendo. A versão 3 foi a última versão de **gated** a estar livre de toda essa problemática e burocracia e funciona bem da mesma forma. A versão 3.6 era o release atual (da versão 3) na época em que este livro foi escrito e é esta a versão que descrevemos a seguir.

gated suporta RIP (ambas as versões), OSPF e IS-IS para roteamento interno e também os protocolos externos EGP e BGP. Um protocolo mais antigo chamado HELLO é suportado por razões históricas.

Início e controle do gated

gated é normalmente iniciado no momento do boot sem nenhum argumento. A maneira correta de fazer isso depende do sistema; consulte o Capítulo 2, *Inicialização e Desligamento*, para maiores informações.

4. **routed** tem uma má reputação por comportamento inadequado em vários sistemas. Um de nossos revisores foi mais longe a ponto de dizer: "Simplesmente não se deve confiar em **routed**."

Snow666

gated obtém suas instruções operacionais por meio de um único arquivo de configuração. O arquivo de configuração normalmente é **/etc/gated.conf**, porém este pode ser alterado por um flag na linha de comando. A partir do momento em que estiver sendo executado, **gated** pode ser manipulado pelo comando **gdc**, que é instalado juntamente com ele. A maioria das aplicações de **gdc** assume a forma

gdc *comando*

Os comandos **gdc** mais comuns são apresentados a seguir:

interface sinaliza **gated** para verificar novamente a lista de interfaces de rede ativas do kernel. **gated** faz isso periodicamente por conta própria, porém se você acabou de alterar a configuração de uma interface, talvez queira forçar uma atualização imediata.

reconfig faz com que **gated** releia o seu arquivo de configuração.

checkconf analisa sintaticamente o arquivo de configuração e verifica a existência de erros de sintaxe, porém não diz a **gated** para carregá-lo.

toggletrace inicia ou interrompe o registro em log.

stop termina **gated**: de maneira cortês, se possível, de maneira forçada, caso contrário.

start gera um **gated** caso um já não esteja em execução.

restart interrompe e reinicializa **gated**. Equivalente a **stop** seguido por **start**.

Tracing

gated pode ser executado com depuração (chamada "rastreio") ativada, fazendo com que suas ações sejam registradas num arquivo de log. Os recursos de rastreio de **gated** são muito úteis quando escrevemos o arquivo de configuração pela primeira vez; eles também fornecem um histórico das tabelas de roteamento.

Dependendo de quais opções de rastreio estiverem habilitadas, o arquivo de log pode crescer rapidamente e deve ser reinicializado ou truncado periodicamente. **gdc toggletrace** fechará completamente o arquivo de log, permitindo que você o renomeie ou o trunque. Um segundo comando **gdc toggletrace** ativa novamente o registro.

As opções de rastreio podem ser especificadas no arquivo de configuração ou na linha de comando **gated** (precedida por **-t** para opções, e por nada para nome do arquivo de log). No arquivo de configuração, várias opções podem ser especificadas para cada protocolo; na linha de comando, eles são globais.

As opções mais úteis são apresentadas a seguir.

all ativa todas as opções de rastreio.

normal rastreia eventos normais. Eventos anormais são sempre rastreados.

policy rastreia a maneira pela qual instruções de políticas configuradas administrativamente afetam a distribuição de rotas.

route rastreia alterações nas tabelas de roteamento.

general ativa tanto **normal** quanto **route**.

Um nível ainda mais detalhado de rastreio faz o dump de pacotes de rede individuais para o arquivo de log. Porém, o rastreio de pacotes pode ser habilitado somente no arquivo de configuração, não na linha de comando.

O arquivo de configuração de gated

Diferentemente de muitos sistemas administrativos Linux, **gated** possui um comportamento razoavelmente padrão. São suportadas centenas de opções, mas redes simples devem precisar somente de algumas poucas linhas de configuração. Enquanto lê o restante deste capítulo e a documentação referente ao comando **gated**, tenha em mente que a maior parte dos recursos não se aplicará ao seu caso.

As próximas seções dão uma visão rápida das principais opções de configuração de **gated**. Devido à maneira como funciona o arquivo de configuração de **gated**, é necessário mostrar descrições de sintaxe para muitas de suas opções. Nós resumimos essas descrições para que contenham somente as partes sobre as quais queremos discursar. Se você estiver pensando que realmente deve haver uma opção XYZ em algum lugar, provavelmente ela existirá; nós apenas não a mostramos. Consulte a documentação de **gated** para obter uma descrição completa.

gated vem com uma explicação do formato de seu arquivo de configuração, porém a documentação não fará muito em termos de ensiná-lo sobre questões avançadas de roteamento. Talvez você precise dar uma olhada na lista de fontes no final do capítulo, para realmente compreender a função e o propósito de cada uma das opções.

Snow666

O arquivo de configuração de **gated** é formado por uma série de instruções separadas por ponto-e-vírgulas. Identificadores são separados por espaços em branco, que podem incluir linhas novas. Chaves são algumas vezes usadas para agrupamento, mas somente em contextos específicos.

Há várias classes de instruções. Instruções de um mesmo tipo têm de aparecer juntas no arquivo de configuração, e as seções têm de estar na seguinte ordem:

- Opções e definições (inclusive declarações de interfaces de rede)
- Configuração de protocolos individuais
- Rotas estáticas
- Controles de importação, exportação e de agregação

Não há nenhum problema de uma seção ser vazia.

As opções de rastreio podem aparecer em qualquer lugar. Se elas estiverem entre chaves, se aplicam somente dentro do contexto da opção ou do protocolo que está sendo configurado. As opções são especificadas através de uma instrução **traceoptions**:

```
traceoptions ["log" [replace] [size size[k|m] files num] ] trace_options
      [except trace_options] ;
```

log é o nome de arquivo no qual a saída do rastreio é armazenada. Se **replace** for especificado, o log será truncado e reinicializado sempre que **gated** for reinicializado; o padrão é anexar. O parâmetro **size** especifica o tamanho máximo do arquivo de log em kilo ou megabytes. Quando o log ficar muito grande, ele será reinicializado e o log antigo renomeado *log*.**1**, *log*.**2** etc., até o número de arquivos especificado pela estrutura **files**. Se especificarmos **size**, também temos de especificar **files**.

As possíveis opções de rastreio (*trace_options*) são aquelas especificadas acima (algumas opções secundárias adicionais não listadas aqui também são suportadas).

Eis um exemplo que cria o arquivo **/var/log/gated.log** e reveza arquivos de 1MB até 3 níveis de profundidade, com todas as opções de rastreio possíveis ativadas:

```
traceoptions "/var/log/gated.log" replace size 1m files 3 all;
```

Instruções de configuração de opções

As opções mais comuns são:

```
options [nosend] [noresolv] [syslog [upto] log_level] ;
```

Os argumentos possuem os seguintes significados:

nosend impede **gated** de enviar quaisquer pacotes. Este argumento é útil para depuração, já que poder ser solicitado a **gated** para processar informações de outros roteadores sem interferência no roteamento em si da rede.

noresolv impede que **gated** tente usar o DNS para converter nomes de hosts em endereços IP. As consultas DNS podem falhar caso não haja informações de roteamento suficientes disponíveis para processá-las. Recomendamos não utilizar nomes de hosts em qualquer contexto de configuração de rede, e esta opção pode ser usada para ajudar a reforçar esta política.

syslog controla o nível de informações que é registrado em log via syslog. Esta opção é sem sentido caso sua instalação não use syslog. Os níveis de log (*log_levels*) válidos são listados na página de manual de **syslog.conf**. O padrão é **syslog upto info**.

Consulte o Capítulo 11 para obter mais informações sobre syslog.

Exemplo:

```
options noresolv;
```

Definições de interfaces de rede

As propriedades das interfaces de rede são configuradas pela instrução **interfaces**, que possui o seguinte formato:

```
interfaces {
   options [strictinterfaces] ;
```

Snow666


```
    define address [broadcast addr] | [pointtopoint addr]
    interface iflst [preference prf] [passive] [simplex] ;
    [netmask mask] [multicast] ;
} ;
```

Pode haver várias instruções **options**, **interface** ou **define** — ou nenhuma. Isso normalmente é verdadeiro para todas as estruturas ao longo do arquivo de configuração **gated**.

A opção **strictinterfaces** torna ilegal referir-se a uma interface no arquivo de configuração que não possa ser localizado pelo kernel no momento da inicialização e que não foi listado numa instrução **define**. Isso sempre é um erro, mas sem **strictinterfaces** estar ativado, ele não é fatal.

Verifique o Capítulo 13 para obter maiores informações sobre links ponto-a-ponto.

A instrução **define** descreve uma interface de rede que pode ou não estar presente na inicialização. Ela é usada mais para links discados ou para outras interfaces que poderiam "surgir" posteriormente. Um outro bom exemplo é uma placa de rede Ethernet PCMCIA que poderia ser inserida num slot algum tempo depois.

Uma instrução **interface** (repetida aqui para fins de clareza)

```
interface iflst [preference prf] [passive] [simplex] ;
```

configura as opções para uma determinada interface ou conjunto de interfaces. *iflst* pode ser o nome de uma interface como **eth0** ou **exp1**, um curinga para nomes como **eth** ou **exp** (atendendo a todas as ocorrências daquele tipo de interface), um nome de host ou endereço IP (indicando a interface à qual aquele endereço está vinculado) ou a string literal **all**.

Se uma interface for **passive**, os roteamentos por meio dela serão mantidos mesmo se aparentemente não conectados de forma apropriada. Se ela for **simplex**, a interface não poderá receber seus próprios pacotes de difusão ampla. **gated** normalmente é capaz de descobrir isto por conta própria, porém em alguns sistemas talvez você tenha de informar explicitamente.

O campo **preference** requer uma explicação mais elaborada. Já que os protocolos de roteamento trabalham de várias maneiras e usam métricas de custo diferentes para determinar os caminhos "mais curtos", não há nenhuma garantia de que quaisquer dois protocolos concordarão sobre quais rotas são as melhores. Quando os resultados computados por protocolos diferentes forem incompatíveis, uma política administrativa terá de ser usada para decidir quais rotas serão propagadas para a tabela de roteamento do kernel e exportadas para outros roteadores.

Para implementar essa política, **gated** associa um valor preferencial numérico a cada rota. Quando duas rotas são conflitantes, aquela com nível de preferência mais baixo será a vencedora do "concurso de beleza" de roteamento.

Os valores de preferência podem se originar de várias fontes. Cada protocolo de roteamento possui um valor de preferência intrínseco. Os valores preferências também podem ser atribuídos a interfaces de rede e a gateways remotos. Já que a maioria das rotas possui um protocolo de origem *e* uma interface de rede *e* um gateway remoto, pode haver vários valores de preferência candidatos. **gated** atribui o mais específico desses valores. Em outras palavras, o caminho com o valor de preferência numérico mais baixo (conforme determinado por qualquer uma das métricas de preferência para o caminho) é o caminho escolhido.

Normalmente, as rotas para redes conectadas diretamente possuem um valor de preferência igual a 0. Se for necessário dar preferência a determinada interface em relação a outra, você poderá usar estruturas **preference** para priorizá-las. Alguns dos demais valores de preferência padrão usados por **gated** são ilustrados na Tabela 14.3.

Tabela 14.3 Valores de preferência de rotas-padrão.

Fonte de informações de roteamento	Preferência
Encaminha para redes conectadas diretamente	0
Rotas reconhecidas via OSPF	10
Redirecionamentos ICMP	30
Rotas estáticas definidas externamente	40
Rotas estáticas definidas em **gated.conf**	60
Rotas reconhecidas por meio de RIP	100
Rotas através de interfaces ponto-a-ponto	110
Rotas através de interfaces que estão inativas	120

Snow666

O exemplo a seguir configura a interface eth0 para ser passiva, significando que ela não será usada para anunciar qualquer informação de roteamento:

```
interfaces {
   interface eth0 passive;
};
```

Outras definições variadas

Vários outros parâmetros globais podem ser configurados na seção de definições.

routerid *host* ;

A instrução **routerid** configura o número de identificação do roteador, que é usado pelos protocolos BGP e OSPF. Ele deve ser listado na forma de um endereço IP e o seu padrão é o endereço da primeira interface física da máquina. Esse valor é importante para os protocolos que o utilizam, e outros roteadores podem vir a precisar fazer referência a ele de forma explícita em seus arquivos de configuração.

```
martians {
host host [allow] ;
network [allow] [exact | refines] ;
network mask mask [allow] [exact | refines] ;
network masklen number [allow] [exact | refines] ;
default [allow] ;
} ;
```

Martian são rotas para destinos que você preferiria ignorar. Pode haver roteadores mal configurados em sua rede que estão transmitindo (difusão ampla) rotas para destinos adulterados ou você poderia simplesmente excluir determinados destinos da tabela de roteamento. Qualquer rota a um destino listado numa instrução **martians** será simplesmente ignorada por **gated**.

Cada destino de rota possui um endereço e uma máscara associada a ele. As diversas variantes da especificação são apenas maneiras diferentes de fornecer um par endereço/máscara em relação aos quais estes podem ser concordantes.

Um número de rede com um **mask** ou **masklen** especifica explicitamente dois valores. Se não for fornecida uma máscara, será usada a máscara implícita pela classe intrínseca do endereço.

exact e **refines** requerem formas diversas de concordância de endereços. Na verdade, há três. Com nenhuma das palavras-chave, a máscara do destino é ignorada. Desde que a parte do endereço de destino coberta pela máscara do critério coincida com o endereço do critério, o destino é considerado como um martian.

Se for especificado **exact**, o endereço e a máscara de destino têm de ser ambos concordantes com os valores determinados pelo critério exatamente na mesma ordem para o destino ser considerado martian. Uma coincidência exata seleciona uma rede, mas não suas sub-redes ou super-redes.

Se **refines** for especificado, a máscara do destino tem de ser mais longa do que a do critério. Se esta for mais longa, então os endereços serão comparados normalmente (usando-se somente a máscara do critério estabelecido). Isso tem o efeito de selecionar as sub-redes de uma rede sem selecionar a rede em si.

As entradas

```
host host ;
default ;
```

são equivalentes a

```
host mask 255.255.255.255 exact ;
0.0.0.0 mask 0.0.0.0 exact ;
```

A palavra-chave **allow** pode ser usada para reabilitar determinados endereços desabilitados por uma especificação mais ampla anterior. Por exemplo:

```
martians {
   128.138.0.0 mask 255.255.0.0 ;
   128.138.145.0 mask 255.255.255.0 allow ;
} ;
```

Essa configuração rejeita todas as informações sobre a rede classe B 128.138. Entretanto, as rotas para a sub-rede 128.138.145 são aceitas. A regra mais específica sempre tem precedência.

Snow666

Configuração de protocolo RIP

Ambas as versões do protocolo RIP são configuradas com uma instrução **rip**.

```
rip yes | no | on | off [ {
   broadcast ;
   nobroadcast ;
   preference pref ;
   defaultmetric metric ;
   interface interface_list
      [noripin | ripin] [noripout | ripout]
      [version 1] | [version 2 [multicast|broadcast] ] ;
   trustedgateways gateway_list ;
   sourcegateways gateway_list ;
   traceoptions [packets | request | response [detail] ] ;
} ] ;
```

yes e **no** são sinônimos de **on** e **off**. O RIP é habilitado por padrão, de modo que você tem de incluir a linha

```
rip no ;
```

caso não queira executar o RIP em si. As opções **broadcast** e **nobroadcast** são similares aos flags **-s** e **-q** de **routed**: **broadcast** faz com que atualizações RIP sejam enviadas mesmo se o host não parecer estar em mais de uma rede. **nobroadcast** impede que RIP envie quaisquer atualizações.

A estrutura **defaultmetric** atribui o custo *metric* às rotas que foram descobertas por meio de outros protocolos quando elas são retransmitidas através de RIP. Essa é uma forma muito bruta de conversão, porém realmente não há uma solução mais refinada. Por padrão, esse valor é configurado em 16 — inatingível — de modo que as rotas de outros protocolos jamais serão transmitidas via RIP. Se quiser redistribuir essas rotas, um bom valor a ser usado é 10.

As opções por interface são configuradas com o mesmo tipo de especificação de interface encontrada na instrução **interfaces** anterior no arquivo de configuração. **ripin** aceita atualizações RIP numa interface, e **noripin** as rejeita. **ripout** e **noripout** são basicamente versões específicas a interfaces de **broadcast** e **nobroadcast**. **noripout** é o padrão em links ponto-a-ponto como os discados.

A instrução **version** informa se deve ser executado RIP-1 ou RIP-2 nas interfaces especificadas. O padrão ao executar RIP-2 é o de transmitir (multidifusão) atualizações em vez de transmiti-las via difusão ampla, o que impede que roteadores RIP-1 as vejam. Podemos especificar **broadcast** para forçar a ocorrência da difusão ampla.

Como padrão, **gated** reconhecerá atualizações RIP de qualquer um que as enviar. Entretanto, se houver uma lista **trustedgateways**, **gated** prestará atenção apenas aos hosts listados. Uma lista de gateways (*gateway_list*) é apenas uma série de endereços IP separados por espaços em branco.

sourcegateways são hosts para os quais atualizações RIP devem ser enviadas diretamente e não via difusão ampla. Esse recurso pode ser usado para atingir hosts em diferentes redes ou para ter como alvo roteadores numa rede que não ofereça suporte à difusão ampla (ou na qual a difusão ampla foi desabilitada).

traceoptions são especificadas conforme descrição da seção 14.3. Quaisquer opções inclusas aqui se aplicarão somente ao RIP. As opções para rastreio de pacotes específicas ao RIP **request**, **response** e **packets** registram em log, respectivamente, solicitações recebidas, respostas enviadas e todos os pacotes. Os pacotes são normalmente sintetizados. Se for especificado **detail**, um dump mais detalhado de cada pacote será incluído no log.

Uma verdadeira estrutura de configuração RIP aparece na seção 1 do exemplo de arquivo de configuração completo chamado **gated.conf**, na seção 14.3.

Algumas informações preliminares sobre OSPF

Antes de mergulharmos nos detalhes sanguinolentos da configuração OSPF (que, na verdade, não são tão sanguinolentos assim), precisamos falar um pouco a respeito de dois recursos adicionais do OSPF: área de roteamento e roteadores designados.

Áreas de roteamento

Numa grande instalação, pode ser que não seja necessário, ou desejável, distribuir um conjunto completo de estados de link de uma ponta à outra da rede. Para reduzir o nível de tráfego de atualização, o OSPF permite que redes individuais sejam agrupadas em "áreas". As informações sobre estado de link (isto é, informações sobre a topologia física da rede) são propagadas somente dentro de uma área; informações sobre a área são distribuídas para o mundo externo na forma de sínteses de roteamento.

www.snow666.blogspot.com
Snow666

Cada rede é membro somente de uma única área, e as áreas podem abranger mais de uma rede. Os roteadores são considerados membros de todas as áreas nas quais eles possuem interfaces de rede. Um roteador pertencente a mais de uma área é denominado roteador de fronteira de área e é responsável pela conversão de registros de estados de link em registros sintetizados.

Uma síntese de roteamento é realmente apenas um conjunto de rotas: "O roteador X é capaz de enviar pacotes à rede Y em 3 saltos" onde X é um roteador de fronteira de área. Os roteadores fora de uma área combinam o custo sintetizado declarado com o custo computado para o roteador de fronteira de área para determinar um custo de caminho total para a rede.

Esse esquema poderia soar como um protocolo de roteamento de vetor de distância disfarçado, porém há duas diferenças importantes. A primeira delas é que as sínteses são propagadas exatamente como chegam do roteador de fronteira de área que as originou. Um roteador poderia concluir que se ele está a dois saltos de distância de X e X a 3 saltos de distância de Y, então ele deve estar a 5 saltos de distância de Y. Contudo, ele jamais revelará o resultado desse cálculo a um outro roteador. Ele simplesmente repassará a rota sintetizada original.[5]

A segunda diferença em relação a um protocolo de vetor de distância é que o esquema OSPF não tenta lidar com as topologias de rede arbitrárias. O OSPF requer que todas as áreas de roteamento sejam logicamente adjacentes a uma área de backbone central (apesar de elas poderem ser adjacentes entre si). Sínteses de rotas podem trafegar somente de uma área de extremidade para o backbone, e vice-versa, e não diretamente entre áreas de extremidades.[6] Essa hierarquia simples de dois níveis impede a possibilidade de criação de loops.

Se a arquitetura efetiva de sua rede não atende ao modelo de duas camadas do OSPF, nem tudo está perdido. Mesmo assim você poderá representá-la como uma hierarquia de duas camadas usando um conceito do OSPF chamado "links virtuais". Infelizmente, uma discussão sobre links virtuais está além do escopo deste livro.

Roteadores designados

Teoricamente, os protocolos de estado de link distribuem informações de roteamento na forma de registros que descrevem a conexão entre roteadores; por exemplo, "O roteador A é adjacente ao roteador B, e o custo do link é 1". Se houvesse 6 roteadores numa rede, provavelmente teriam de ser propagados 30 anúncios de estado de link diferentes, pois cada roteador era adjacente a 5 outros roteadores. Pior ainda, os roteadores teriam de tratar cada um dos demais como vizinhos e certificarem-se de que seus bancos de dados estariam sincronizados.

O OSPF reduz a quantidade de informações que é propagada apontando um dos roteadores numa rede como sendo o *roteador designado*.[7] O roteador designado escuta todos os anúncios de estado de link de todos os demais roteadores e então envia um resumo do que ele tomou conhecimento.

Os roteadores de uma rede cooperam para eleger um roteador designado tomando como base um valor de prioridade por roteador que é configurado administrativamente. Roteadores com um valor de prioridade 0 são inelegíveis para roteador designado. Entre os roteadores remanescentes, aquele com a maior prioridade é declarado vencedor. Se houver empate, um roteador é selecionado de modo pseudo-aleatório.

Um roteador designado de backup também é selecionado da mesma maneira. Cada roteador da rede mantém um relacionamento contínuo tanto com o roteador designado quanto com o backup do roteador designado. Se o roteador designado falhar, o backup entra imediatamente em ação para assumir o controle, e um novo backup é eleito.

Os roteadores designados manipulam uma quantidade de tráfego um pouco maior do que seus associados, de forma que eles devem ser selecionados apropriadamente.

Configuração de protocolo para o OSPF

Em **gated.conf**, configuramos opções OSPF por meio da instrução **ospf**:

```
ospf yes | no | on | off [ {
   defaults {
      router-prio ;
   } ;
   traceoptions trace_options ;
```

5. Isso é verdadeiro somente dentro de uma área. Se o registro de síntese atravessar para uma área adjacente, o roteador de fronteira de área que o encaminha reafirmará as informações relativas a si próprio.
6. OK, essa afirmação não é inteiramente verdadeira. Há alguns poucos tipos de áreas especiais, como a NSSA ("Not So Stubby Area", ou seja, "áreas não tão grosseiras") que poderão passar informações de roteamento entre redes ramais, porém cada rota aprendida desta maneira é marcada de forma especial de modo que ela não crie um loop acidentalmente.
7. Na verdade, "domínio de difusão ampla" seria um termo melhor do que "rede" nesse caso. Há também um mecanismo definido para suportar o conceito de um roteador designado ou uma rede que não use difusão ampla (mas sim multiacesso).

www.snow666.blogspot.com
Snow666


```
   backbone | (area area) {
      networks {
         network [exact | refines] [restrict] ;
         network mask mask [exact | refines] [restrict] ;
         network masklen num [exact | refines] [restrict] ;
         host host [exact | refines] [restrict] ;
      } ;
      stubhosts {
         host cost cost ;
      } ;
      interface interface_list [cost cost] {
         enable | disable ;
         priority priority ;
      } ;
   } ;
} ] ;
```

Essa instrução é menos complicada do que parece. **on**, **off**, **yes** e **no** possuem significados óbvios; o padrão é não executar OSPF.

Na seção **defaults**, **router-prio** especifica uma prioridade-padrão do roteador (a ser usada ao eleger um roteador designado) igual a 1 em todas as interfaces. Esse valor pode ser alterado em interfaces específicas, se desejado. A prioridade-padrão é 0, que torna **gated** inelegível para se tornar o roteador designado de qualquer rede.

A definição de cada área começa com a palavra-chave **backbone** ou então **area**. Deve haver uma instrução **area** ou de **backbone** para cada área da qual o roteador é um membro. O backbone é definido como área 0 na especificação de protocolo OSPF, porém **gated** requer que usemos a palavra-chave **backbone** em vez de **area 0**.

area pode ser dado na forma de um número decimal ou então de um número de quatro bytes no formato de endereço IP (por exemplo, 128.138.45.2). O número de área jamais é interpretado como um endereço IP por **gated**, contudo o formato usando quatro blocos separados por pontos é suportado no caso de você querer rotular áreas com os endereços IP dos principais roteadores ou servidores.

A lista de redes (**networks**) define as redes que compõem a área. Somente é necessário enumerar redes nos arquivos de configuração dos roteadores de fronteira de área. A especificação de endereços e máscaras é feita exatamente como para **martians**, conforme descrito na seção 14.3, exceto que não há nenhuma palavra-chave **allow**. Redes marcadas com a opção **restrict** não são inclusas nas sínteses de rotas; elas são redes "secretas" atingíveis somente dentro da área.

stubhosts são hosts conectados diretamente que devem ser anunciados como atingíveis via este roteador com o custo (*cost*) especificado (1 normalmente é um bom número). Esse recurso é usado basicamente para suportar links a hosts que estão conectados via PPP ou SLIP.

Finalmente, a lista de interfaces **interface** especifica o custo (*cost*) para as redes conectadas (normalmente 1, o padrão) e a prioridade (*priority*) desse **gated** em vir a se tornar o roteador designado. Se uma interface for configurada em **disable**, não ocorrerá nenhuma conversação OSPF através desta interface.

Um exemplo real de uma estrutura de configuração OSPF pode ser visto na seção 2 do arquivo de configuração completo, **gated.conf**, na página 245.

Configuração de protocolo para redirecionamentos ICMP

gated permite que exerçamos algum controle administrativo sobre a manipulação de rotas aprendidas por meio de redirecionamentos ICMP (veja a página 197 para uma explicação sobre o seu significado).

```
redirect yes | no | on | off [ {
   preference preference ;
   interface interface_list [noredirects] | [redirects] ;
   trustedgateways gateway_list ;
   traceoptions trace_options ;
} ] ;
```

Todas essas opções agora já devem lhe ser familiares. **preference** configura a preferência de rotas para rotas derivadas de redirecionamento em geral (o padrão é 30, que é relativamente bom). As opções **redirects** e **noredirects** habilitam e desabilitam a aceitação de redirecionamentos por interface de rede e **trustedgateways** os habilitam somente quando enviados por roteadores específicos.

Snow666

Não há nenhuma opção de rastreio específico de redirecionamento.

Rotas estáticas

As rotas estáticas são configuradas pela instrução **static**:

```
static {
      dest gateway gateway_list [interface interface_list] [preference preference]
         [retain] [reject] [blackhole] [noinstall] ;
} ;
```

dest pode ser especificado com qualquer um dos suspeitos usuais:

```
host host
default
network
network mask mask
network masklen length
```

gateway_list é o conjunto de roteadores por meio do qual esse destino pode ser atingido. Se o gateway designado não estiver numa rede conectada diretamente (via uma das interfaces especificadas na lista de interfaces (*interface_list*) opcional, a rota será ignorada.

A preferência de rotas tem como padrão 60, que permite que ela seja sobreposta por rotas computadas OSPF ou por redirecionamentos ICMP.

Se uma rota é marcada com **retain**, ela será deixada na tabela de roteamento do kernel quando **gated** sair. Normalmente, **gated** limpa o que ele fez e deixa somente as rotas pré-existentes e de interfaces. Inversamente, a opção **noinstall** faz com que a rota não seja instalada na tabela de roteamento local, mas somente fique disponível para propagação para outros roteadores. Essa opção é útil em roteadores que agem como "servidores de rotas", significando que eles realmente não direcionam tráfego, mas sim coordenam informações de roteamento para a infra-estrutura de rede (eles podem ter acesso "somente para gerenciamento de tráfego" a uma rede alternativa que transporta seus pacotes).

Rotas marcadas com as marcas **blackhole** e **reject** impedem a ocorrência de encaminhamento em sistemas que suportam estes recursos. Com **reject**, um erro ICMP é retornado para o emitente; com **blackhole**, os pacotes apenas desaparecem misteriosamente sem deixar vestígios, um tanto parecido com o e-mail da Evi.

Um exemplo de rota estática é apresentado na seção 3 do modelo completo de arquivo de configuração, **gated.conf**, na Seção 14.5.

Rotas exportadas

Assim que **gated** tiver computado as rotas que ele deseja, o padrão é que as coloque na tabela de encaminhamento do kernel. Para a maioria dos aplicativos, isso é tudo o que é necessário. Algumas vezes, entretanto, é desejável configurar **gated** para atuar como uma espécie de tradutor, aceitando informações de um protocolo e as distribuindo num outro. Isso é feito no arquivo de configuração por meio da estrutura **export**:

```
export proto protocol
   [interface interface_list | gateway gateway_list]
   restrict ;
ou
   export proto protocol
      [interface interface_list | gateway gateway_list]
      [metric metric] {
         export_list ;
      } ;
```

Nesse caso, *protocol* é o protocolo de roteamento que estará anunciando as informações traduzidas, e *export_list* é o que deve ser traduzido, conforme especificado, listando uma estrutura **proto** para cada conjunto de dados a ser traduzido. Eis um exemplo de *export_list*:

```
proto static {
   ALL metric 1;
} ;
```

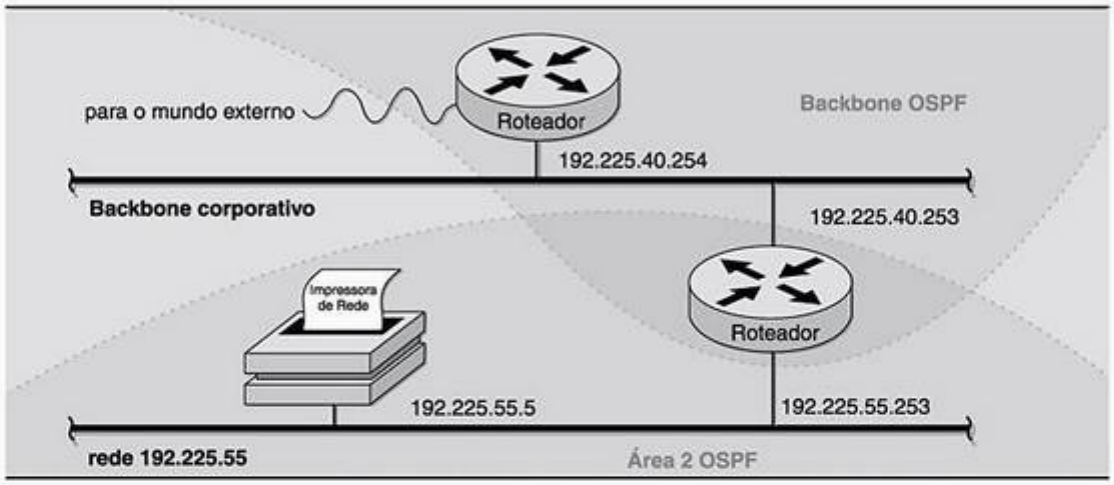
Snow666

Esse pequeno trecho de código converte todas as rotas estáticas e as insere na lista exportada com uma métrica igual a 1.

Um exemplo de configuração gated completo

A configuração a seguir destina-se a um ambiente no qual tanto RIP quanto OSPF estão em uso. A configuração é para o roteador de fronteira de área mostrado no Diagrama C.

Diagrama C Topologia de rede para nosso exemplo de configuração gated



A rede superior (o backbone corporativo) foi padronizada em OSPF, porém a LAN abaixo possui dispositivos (algumas poucas impressoras de rede) capazes de reconhecer somente RIP. Neste ambiente, **gated** é usado para retransmitir (difusão ampla) as rotas OSPF via RIP. Esta é uma boa configuração para um ambiente corporativo ou de campus, pois ela permite que PCs e dispositivos de rede tomem conhecimento de informações via RIP, embora use um protocolo mais sensível para se comunicar no backbone entre grupos, andares e prédios. O conteúdo do arquivo de configuração é apresentado a seguir:

Section 1:

```
rip yes {
      broadcast;
      defaultmetric 10;
      interface 192.225.40.253 noripout;
      interface 192.225.55.253 ripout;
};
```

Section 2:

```
ospf yes {
      area 0.0.0.2 {
         authtype none;
         networks {
            192.225.55.0 mask 255.255.255.0;
         };
         interface 192.225.55.253 cost 1 {
            priority 2;
         };
   };

   backbone {
         interface 192.225.40.253 {
            priority 2;
         };
      };
   };
```

Snow666

Section 3:

```
static {
        default gateway 192.225.40.254 preference 140 retain;
};
```

Section 4:

```
export proto rip {
        proto ospf {
           ALL metric 1;
        };
        proto direct {
           ALL metric 1;
        };
        proto static {
           ALL metric 1;
        } ;
};
```

Section 5:

```
export proto ospf {
        proto direct {
           ALL metric 1;
        };
};
```

Na seção 1, **gated** é instruído a falar o protocolo RIP. Ele escuta transmissões RIP de outros roteadores em ambas as interfaces, mas envia pacotes RIP próprios somente por meio da interface 192.225.55.253. Essa restrição serve para eliminar tráfego de difusão ampla no backbone corporativo.

A seção 2 habilita o OSPF. A interface 192.225.40.253 se encontra na área 0, a área de backbone. Ela enviará mensagens OSPF HELLO para outros roteadores nessa rede para descobrir quem são seus vizinhos. 192.225.55.253 se encontra na área 2. (Pode haver ou não outros roteadores abaixo que queiram compartilhar informações via OSPF.)

Atualmente, há somente um único caminho dessa rede para o mundo externo. Portanto, como um adicional, a seção 3 inclui uma rota-padrão estática para o gateway da Internet na rede 192.225.40.0.

Nas seções 4 e 5, informamos a **gated** quais rotas devem ser anunciadas, respectivamente, via RIP e OSPF. Queremos que os anúncios RIP incluam quaisquer redes conectadas diretamente, a rota-padrão estática e quaisquer rotas aprendidas via OSPF. Queremos que os anúncios OSPF incluam quaisquer redes conectadas diretamente (como, por exemplo, a rede 192.225.55.0). Já que somos um roteador interno, não queremos anunciar uma rota-padrão.

14.6 Critérios de Seleção de Estratégia de Roteamento

Há, basicamente, quatro níveis de complexidade nos quais o roteamento para uma rede pode ser gerenciado:

- Nenhum roteamento
- Somente rotas estáticas
- A maioria das rotas estáticas, porém os clientes ouvem atualizações RIP
- Roteamento dinâmico em toda parte

A topologia da rede como um todo possui um efeito dramático nas necessidades de roteamento de cada segmento individual. Redes diferentes podem precisar de níveis muito diversos de suporte a roteamento. As regras práticas a seguir podem ajudá-lo a escolher uma estratégia:

- Uma rede autocontida (isolada) não requer nenhum roteamento.
- Se houver somente um caminho para fora de uma rede, os clientes (máquinas não-gateway) dessa rede devem ter uma rota-padrão estática para o único gateway. Não é necessária nenhuma configuração adicional, exceto no gateway em si.
- Um gateway com um pequeno número de redes de um lado e um gateway para "o mundo" no outro lado pode ter rotas estáticas explícitas apontando para o primeiro e uma rota-padrão para o último. Entretanto, o roteamento dinâmico é aconselhável caso haja mais de uma opção de roteamento em ambos os lados.
- Se você usa RIP e está preocupado com a carga de sistema e de rede que isto acarreta, evite usar **routed** no modo ativo — ele transmite tudo o que conhece (correto ou não) em pequenos intervalos. **gated** permite que especifiquemos quais rotas devem ser enviadas ("anunciadas") reduzindo, portanto, o excesso de infor-

www.snow666.blogspot.com
Snow666

mações de roteamento. **gated** também pode enviar atualizações RIP para determinados gateways em vez de transmiti-las para todos os pontos.

- Para que os clientes ouçam passivamente atualizações de roteamento sem enviar suas próprias informações, use **routed -q**. Os clientes também podem ouvir passivamente com **gated**, porém ele é bem mais pesado.
- Muitas pessoas irão lhe informar que RIP é um protocolo horrível e terrível e que **routed** é criação do diabo. Não é necessariamente assim. Se ele funciona e você está contente com o desempenho, vá em frente e use-o. Você não ganha nada investindo tempo numa estratégia de roteamento excessivamente elaborada.
- Se RIP não for o seu principal protocolo de roteamento, poderá fazer com que **gated** transmita suas informações de roteamento como RIP, puramente para benefício dos clientes passivos.
- **routed** escuta todo o mundo e acredita em tudo que ouve. **gated** lhe dá maior controle sobre as atualizações. Mesmo se sua instalação usar RIP, talvez você queira controlar o intercâmbio de dados de roteamento com **gated** e executar **routed** somente em máquinas-cliente.
- Use roteamento dinâmico em pontos onde as redes cruzam fronteiras políticas ou administrativas.
- Em redes roteadas dinamicamente contendo loops ou caminhos redundantes, use OSPF se possível.
- Roteadores conectados à Internet com vários provedores superiores têm de usar BGP. A maioria dos roteadores conectados à Internet possui somente um caminho superior e, conseqüentemente, uma única rota estática explícita é suficiente.

Uma boa estratégia de roteamento para instalações de porte médio com estrutura local relativamente estável e conexão com a rede de qualquer outro usuário é usar uma combinação de roteamento dinâmico e estático. Máquinas dentro da estrutura local que não possuem um gateway para redes externas podem usar roteamento estático, encaminhando todos os pacotes desconhecidos para uma máquina-padrão que compreenda o mundo externo e realize o roteamento dinâmico.

Uma rede que é muito complicada de ser gerenciada com esse esquema deveria fiar-se no roteamento dinâmico. Rotas estáticas padrão ainda podem ser usadas em redes das extremidades, porém máquinas em redes com mais de um roteador devem executar **routed** no modo passivo. Todas as máquinas com mais de uma interface de rede devem executar **gated** no modo ativo e transmitir rotas com RIP.

14.7 Roteadores Cisco

Os roteadores produzidos pela Cisco Systems, Inc., são um padrão de fato para roteamento Internet hoje em dia. Tendo capturado mais de 70% do mercado de roteadores, os produtos da Cisco são bem conhecidos e pessoal que saiba operá-los é relativamente fácil de ser encontrado. Antes da Cisco, os computadores UNIX com várias interfaces de rede eram freqüentemente usados como roteadores. Hoje, os roteadores são o mecanismo preferido para colocação em gabinetes para comunicação de dados e acima de forros onde os cabos se juntam. Eles são mais baratos, mais rápidos e mais seguros do que seus equivalentes UNIX ou Linux.

A maioria dos produtos para roteamento da Cisco roda num sistema operacional chamado Cisco IOS, que é proprietário e não relacionado com o Linux. Seu conjunto de comandos é bem grande; a documentação completa chega a ocupar 2 m de espaço em prateleira. Jamais poderíamos cobrir completamente o Cisco IOS aqui, porém conhecer alguns fundamentos pode levá-lo para bem longe.

O IOS define dois níveis de acesso (usuário e privilegiado), ambos os quais são protegidos por senha. Como padrão, podemos simplesmente enviar um **telnet** para um roteador Cisco para entrar no modo usuário.[8] Será solicitada a senha de acesso em nível de usuário:

```
% telnet acme-gw.acme.com
Connected to acme-gw.acme.com.
Escape character is '^]'.

User Access Verification
Password:
```

Após introduzir a senha correta, você receberá um prompt do interpretador de comandos EXEC da Cisco.

```
acme-gw.acme.com>
```

8. Pode ser configurada uma série de métodos de acesso. Se sua instalação já utiliza roteadores Cisco, entre em contato com o administrador da sua rede para descobrir que métodos foram habilitados.

www.snow666.blogspot.com
Snow666

Nesse prompt, podemos introduzir comandos como **show interfaces** para ver as interfaces de rede do roteador ou **show?** para obter ajuda sobre as demais coisas que podem ser vistas.

Para entrar no modo privilegiado, digite **enable** e introduza a senha privilegiada quando ela for solicitada. Assim que tiver atingido o nível privilegiado, seu prompt terminará num #:

```
acme-gw.acme.com#
```

TOME CUIDADO — você poderá fazer qualquer coisa com esse prompt, inclusive apagar as informações de configuração do roteador e seu sistema operacional. Quando estiver em dúvida, consulte os manuais da Cisco ou um dos abrangentes livros publicados pela Cisco Press.

Você poderá digitar **show running** para ver a configuração atual em execução do roteador e **show config** para visualizar a configuração não volátil atual. A maior parte do tempo, esses dados são iguais. Eis uma configuração típica:

```
acme-gw.acme.com# show running
Current configuration:
version 12.1
hostname acme-gw
enable secret xxxxxxxx
ip subnet-zero

interface Ethernet0
description Acme internal network
ip address 192.108.21.254 255.255.255.0
no ip directed-broadcast
interface Ethernet1
description Acme backbone network
ip address 192.225.33.254 255.255.255.0
no ip directed-broadcast

ip classless
line con 0
transport input none

line aux 0
transport input telnet
line vty 0 4
password xxxxxxxx
login

end
```

A configuração do roteador pode ser modificada de várias maneiras. A Cisco oferece ferramentas gráficas que são executadas em algumas versões do UNIX/Linux e do Windows. Verdadeiros administradores de redes jamais as usam; o prompt de comando é sempre a "aposta segura". Também é possível enviar um arquivo de configuração via **tftp** de ou para um roteador para você poder editá-lo em seu editor preferido.[9]

Para modificar a configuração por meio do prompt de comando, digite **config term**:

```
acme-gw.acme.com# config term
Enter configuration commands, one per line. End with CNTL/Z.
acme-gw(config)#
```

A seguir, você pode digitar novos comandos de configuração exatamente como quer que eles apareçam na saída de **show running**. Se, por exemplo, quiséssemos alterar o endereço IP da interface Ethernet0 do exemplo acima, poderíamos digitar o seguinte:

```
interface Ethernet0
ip address 192.225.40.253 255.255.255.0
```

Ao terminar de introduzir comandos de configuração, pressione <Control-Z> para retornar para o prompt de comando normal. Se estiver contente com a nova configuração, digite **write mem** para salvar a configuração para memória não volátil.

9. Dica quente: o Microsoft Word não é a melhor opção para esta aplicação.

Snow666

Eis algumas dicas para ser bem-sucedido no uso de roteadores Cisco:

- Atribua um nome ao roteador por meio do comando **hostname**. Esta precaução ajuda a evitar acidentes provocados alterando a configuração no roteador errado. O nome do host sempre aparecerá no prompt de comandos.
- Sempre mantenha uma configuração de roteador de backup em mãos. Você pode criar um pequeno script **expect** que envia via **tftp** a configuração em execução para um computador Linux a cada noite para segurança.
- Controle o acesso à linha de comandos do roteador colocando listas de acesso nas VTYs do roteador (VTYs são como PTYs num computador Linux). Esta precaução evita que pessoal não autorizado tente invadir o seu roteador.[10]
- Controle o tráfego que flui entre suas redes (e, possivelmente, para o mundo externo) com listas de acesso em cada interface. Consulte o Capítulo 21, *Firewalls para Filtragem de Pacotes* para obter maiores informações sobre como configurar listas de acesso.
- Mantenha os roteadores fisicamente seguros. É fácil reinicializar a senha privilegiada se você tiver acesso físico a um equipamento Cisco.

14.8 Leitura Recomendada

Huitema, Christian. *Routing in the Internet. 2. ed. Prentice Hall, 2000.*
Este livro é uma introdução clara e bem escrita sobre roteamento, partindo dos princípios básicos e avançando. Ele cobre a maioria dos protocolos mais comumente usados e também alguns tópicos avançados como multidifusão. Clientes da Amazon.com que compraram este livro também compraram o álbum *The Dirty Boogie* da The Brian Setzer Orchestra.
Moy, John T. *OSPF: Anatomy of an Internet Routing Protocol*. Addison-Wesley, 1998.
Uma exposição completa sobre o OSPF pelo autor do padrão do protocolo OSPF e uma boa dose sobre o **gated**. Clientes da Amazon.com que compraram este livro também compraram o álbum *Stunt* da Barenaked Ladies.
Halabi, Bassam. *Internet Routing Architectures*. 2. ed. Cisco Press, 2000.
Este livro se concentra no BGP, o protocolo de gateway externo mais usado. Clientes da Amazon. com que compraram este livro também compraram a trilha sonora do filme *The Matrix*, contendo as letras das músicas.
Stewart, John W. *BGP4 Inter-domain Routing in the Internet*. Addison-Wesley, 1999.
Trata-se de uma exposição muito prática do BGP4 e de suas aplicações. Clientes da Amazon.com que compraram este livro somente compraram outros livros sobre roteamento e a maioria deles chata.
Há muitas RFCs relacionadas a roteamento. As principais são mostradas na Tabela 14.4.

Tabela 14.4 RFCs relacionadas com roteamento.

RFC	Título	Autores
2328	OSPF Version 2	John T. Moy
1058	Routing Information Protocol	C. Hedrick
2453	RIP Version 2	Gary Scott Malkin
1256	ICMP Router Discovery Messages	Stephen E. Deering
1142	OSI IS-IS Intra-domain Routing Protocol	David R. Oran
1075	Distance Vector Multicast Routing Protocol	D. Waitzman et al.
1519	CIDR: an Address Assignment and Aggregation Strategy	Vince Fuller et al.
1771	A Border Gateway Protocol 4 (BGP-4)	Yakov Rekhter et al.

10. As versões mais recentes do IOS suportam o protocolo SSH. Você deve usar este, em vez da interface TELNET padrão, se ele estiver disponível em seu ambiente.

Snow666

14.9 EXERCÍCIOS

E14.1 Investigue o comando **route** do Linux e escreva uma breve descrição sobre o que ele faz. Usando **route**, como você:

a) Adicionaria uma rota-padrão para 128.138.129.1 usando a interface eth1?
b) Eliminaria uma rota para 128.138.129.1?
c) Determinaria se uma rota foi adicionada por um programa como **gated** ou um redirecionamento ICMP? (Note que este método também funcionará com a saída de **netstat -rn**.)

E14.2 Comparando o roteamento estático com o roteamento dinâmico, liste várias vantagens e desvantagens de cada um deles. Descreva situações nas quais cada um deles seria apropriado e explique o porquê.

★ **E14.3** Considere a saída gerada por **netstat -rn** abaixo. Descreva as rotas e descubra a configuração da rede. Qual rede, 10.0.0.0 ou 10.1.1.0, está mais próxima da Internet? Que processo adicionou cada rota?

```
Destination   Gateway    Genmask        Flags   MSS  Window  irtt  Iface
10.0.0.0      0.0.0.0    255.255.255.0  U        40  0          0  eth1
10.1.1.0      0.0.0.0    255.255.255.0  U        40  0          0  eth0
0.0.0.0       10.0.0.1   0.0.0.0        UG       40  0          0  eth1
```

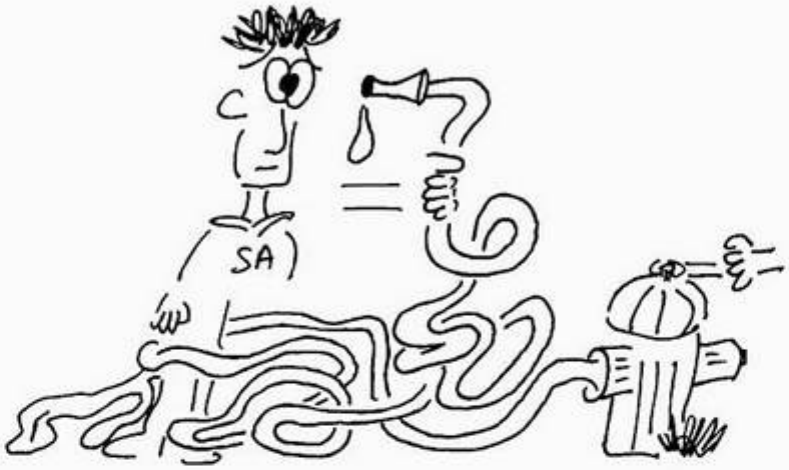
★ **E14.4** O que são as rotas martian em **gated** e por que a definição de martian é útil?

★★ **E14.5** Descubra o esquema de roteamento que é usado em sua instalação. Quais protocolos estão sendo usados? Que máquinas se conectam à Internet? Você pode usar **tcpdump** para procurar pacotes de atualização de roteamento na rede local e **traceroute** para explorar além da rede local. (Requer acesso root.)

★★ **E14.6** Se você fosse um ISP de porte médio que fornecesse contas discadas e hospedagem virtual, que tipo de configuração de roteamento você usaria? Certifique-se de considerar não somente o(s) roteador(es) do gateway entre o backbone Internet e sua própria rede, como também quaisquer roteadores internos que possam estar em uso. Desenhe um diagrama da rede que descreva sua arquitetura de roteamento.

Snow666

15 Hardware de Rede



Nada está influenciando mais a nossa cultura hoje do que a habilidade de transferir grandes quantidades de dados de um local a outro de forma muito rápida. Agora temos conectividade mundial num nível que somente os mais fanáticos por ficção científica poderiam ter sonhado há alguns anos. Por trás de toda esta loucura temos o hardware de rede elaborado e — adivinhe! — uma quantidade enorme de coisas que se originou nas profundas e escuras cavernas do UNIX.

Acompanhar todo este avanço em ritmo alucinante é um desafio. A velocidade e a confiabilidade de uma rede tem efeito direto sobre a produtividade de uma empresa. Uma rede mal projetada é um embaraço pessoal e profissional. Pode até ser muito caro corrigi-la.

Pelo menos três fatores principais contribuem para uma instalação bem-sucedida:

- Desenvolvimento de um projeto de rede razoável
- Seleção de hardware de alta qualidade
- Documentação e instalação apropriadas

As primeiras seções deste capítulo discutem as mídias usadas comumente para redes locais e de longa distância, inclusive Ethernet, ATM, frame relay e DSL. A seguir cobrimos questões de projeto com as quais provavelmente você irá se deparar em qualquer rede, seja ela nova ou antiga.

15.1 LAN, WAN OU MAN?

Temos sorte, num certo ponto, que o TCP/IP pode ser transportado facilmente em uma série de mídias. Na realidade, entretanto, o mercado de hardware de rede é dividido numa série de classificações confusas.

As redes que existem dentro de um prédio ou grupo de prédios geralmente são conhecidas como LANs (Local Area Networks, redes locais). Prevalecem conexões de baixo custo e alta velocidade. As WANs (Wide Area Networks, redes de longa distância) são redes nas quais as extremidades estão dispersas geograficamente, talvez separadas por milhares de quilômetros. Nessas redes, alta velocidade normalmente implica um alto custo, porém praticamente não há limites para as instalações que podemos incluir na rede (Bruges, Bélgica a Sitka, Alasca!). MAN é um termo de marketing de telecomunicações para Metropolitan Area Network, significando uma mídia de custo moderado e de alta velocidade usada dentro de uma cidade ou de um grupo de cidades. Neste capítulo, exploramos algumas das tecnologias usadas para implementar essas feras.

Snow666

15.2 Ethernet: A Lan Comum

Tendo capturado mais de 80% do mercado mundial de LANs, a Ethernet pode ser encontrada praticamente em toda parte em suas várias formas. Ela começou como tese de Ph.D. de Bob Metcalfe no MIT. Bob formou-se e foi para o Xerox PARC; junto com a DEC e a Intel, a Xerox acabou por transformar a Ethernet num produto. Ela foi um dos primeiros casos em que empresas de informática concorrentes aliaram forças num projeto técnico.[1]

O padrão Ethernet foi originalmente especificado para uma velocidade de 3 Mb/s (mega*bits* por segundo), porém passou para 10Mb/s quase imediatamente. Ele foi desenvolvido na Xerox Alto, que não tinha espaço suficiente na placa de circuito impresso para um relógio externo. A interface Ethernet tinha de usar o relógio da Alto, o que significava que a velocidade da rede tinha de ser de 2.94 Mb/s. Esta foi arredondada para 3 Mb/s. Metcalfe e outros desenvolvedores pioneiros que já haviam trabalhado na arquitetura da ARPANET se opuseram a um erro de arredondamento que excedia a largura de banda inteira da ARPANET, contudo o pessoal de marketing venceu.

A Ethernet passou por dificuldades em seus primeiros anos na década de 80, uma época em que uma série de sistemas operacionais de rede, inclusive o UNIX, também estava ganhando conhecimentos básicos de rede aprendendo a lidar uns com os outros. Quando a Ethernet atingiu sua adolescência, ela estava pronta para dominar. Em 1994, a Ethernet chamou a atenção e foi padronizado para a velocidade de 100 Mb/s. Logo após completar 19 anos em 1998, ela estava pronta para enfrentar uma nova guerra — a 1 Gb/s. Agora já adulta com cerca de 25 anos, a Ethernet está caminhando para o novo limite de 10 Gb/s, tendo ofuscado todas as suas rivais.[2] A Tabela 15.1 destaca a evolução dos diversos padrões Ethernet.[3]

Funcionamento da Ethernet

A Ethernet pode ser descrita como um jantar refinado em que os convidados (computadores) não interrompem uns aos outros, mas sim aguardam uma calmaria na conversação (nenhum tráfego na rede) antes de falarem. Se dois convidados começarem a conversar ao mesmo tempo (uma colisão), ambos param, se desculpam, aguardam um pouco e depois um deles começa a falar novamente.

Tabela 15.1 A evolução da Ethernet.

Ano	Velocidade	Nome comum	IEEE#	Distância	Mídia
1973	3 Mb/s	Xerox Ethernet	–	?	Coaxial
1980	10 Mb/s	Ethernet 1	–	500m	Coaxial RG-11
1982	10 Mb/s	DIX Ethernet (Ethernet II)	–	500m	Coaxial RG-11
1985	10 Mb/s	10Base5 ("Thicknet")	802.3	500m	Coaxial RG-11
1985	10 Mb/s	10Base2 ("Thinnet")	802.3	180m	Coaxial RG-58
1989	10 Mb/s	10BaseT	802.3	100m	Cobre UTP[a] categoria 3
1993	10 Mb/s	10BaseF	802.3	2km	Fibra MM[b]
				25km	Fibra SM
1994	100 Mb/s	100BaseTX ("100 meg")	802.3u	100m	Cobre UTP categoria 5
1994	100 Mb/s	100BaseFX	802.3u	2km	Fibra MM
				20km	Fibra SM
1998	1 Gb/s	1000BaseSX	802.3z	260m	Fibra MM 62.5-µm
				550m	Fibra MM 50-µm
1998	1 Gb/s	1000BaseLX	802.3z	440m	Fibra MM 62.5-µm
				550m	Fibra MM 50-µm
				3km	Fibra SM
1998	1 Gb/s	1000BaseCX	802.3z	25m	Twinax
1999	1 Gb/s	1000BaseT ("Gigabit")	802.3ab	100m	Cobre UTP cat 5E e 6
2001	10 Gb/s	10GbE ("10 Gigabit")	802.3ae	65m	Fibra MM
				40km	Fibra SM

a. Par trançado não blindado
b. Fibra em modo simples e multimodo

1. Bob Metcalfe também articulou a "Lei de Metcalfe", que afirma que o valor da rede se expande exponencialmente à medida que o número de usuários aumenta.
2. A especulação atual prevê a disponibilidade em larga escala da Ethernet na casa dos terabits por volta de 2008!
3. Omitimos alguns padrões Ethernet que não se tornaram populares, como o 100BaseT4 e o 100BaseVG-AnyLAN.

Snow666

O termo técnico para esse esquema é CSMA/CD:

- Carrier Sense: você pode dizer se alguém está conversando.
- Multiple Access: todo o mundo pode conversar.
- Collision Detection: você sabe quando interrompe alguém.

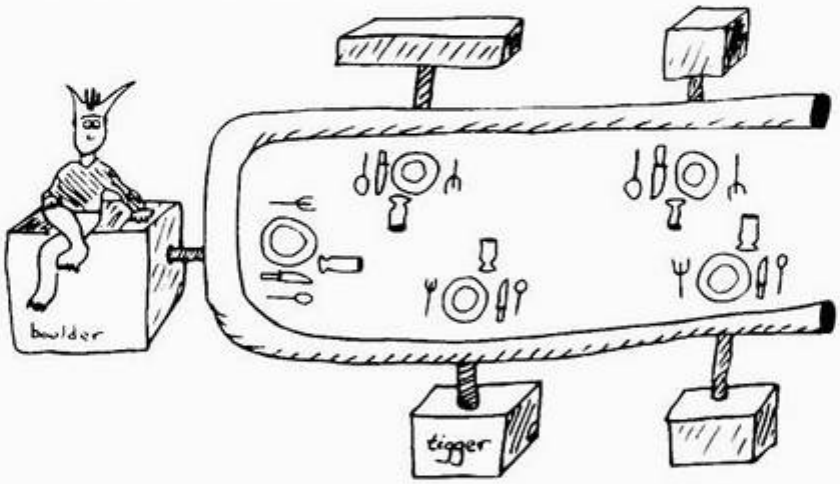
O atraso real após a detecção de colisão é um tanto aleatório. Essa convenção impede a situação na qual dois hosts transmitam simultaneamente para a rede, detectem a colisão, aguardem a mesma quantidade de tempo e depois comecem a transmitir novamente, inundando, conseqüentemente, a rede com colisões. Isso nem sempre é verdade!

Topologia Ethernet

A topologia Ethernet é um barramento ramificado sem loops; há somente uma maneira para um pacote percorrer um caminho entre dois hosts quaisquer da mesma rede. A Ethernet fornece um mecanismo para intercambiar três tipos de pacotes num segmento: unidifusão, multidifusão e difusão ampla. Pacotes unidifusão são destinados somente para um único host. Os pacotes multidifusão são destinados a um grupo de hosts. Os pacotes de difusão ampla são entregues a todos os hosts de um segmento.

Um "domínio de difusão ampla" é o conjunto de hosts que recebem pacotes destinados para o endereço de difusão ampla e há exatamente um domínio de difusão ampla para cada segmento Ethernet. No padrão e mídias Ethernet (como 10Base5) usado logo no início, os segmentos físicos e lógicos eram exatamente os mesmos, já que todos os pacotes trafegavam num único grande cabo com as interfaces de hosts conectadas na lateral dele.[4]

Diagrama A Um "jantar" Ethernet com convidados refinados



Com o advento dos switches (comutadores) modernos, os segmentos lógicos atuais são formados por muitos (possivelmente dezenas ou centenas) de segmentos físicos aos quais somente dois dispositivos estão conectados: a porta do switch e o host. Os switches são responsáveis por acompanharem pacotes multidifusão e unidifusão até os segmentos físicos nos quais os destinatários pretendidos residem; o tráfego de difusão ampla é encaminhado para todas as portas num segmento lógico.

Um único segmento lógico pode ser formado de segmentos físicos operando em velocidades diversas (10 Mb/s, 100 Mb/s, 1 Gb/s ou 10 Gb/s); portanto, os switches têm de ter recursos de buffering[5] e de temporização para eliminar possíveis conflitos.

UTP

O UTP (Unshielded Twisted Pair, par trançado não blindado) é o tipo de mídia de cabo preferida para o padrão Ethernet. Ele se baseia numa topologia estrela e possui várias vantagens em relação a outras mídias:

---

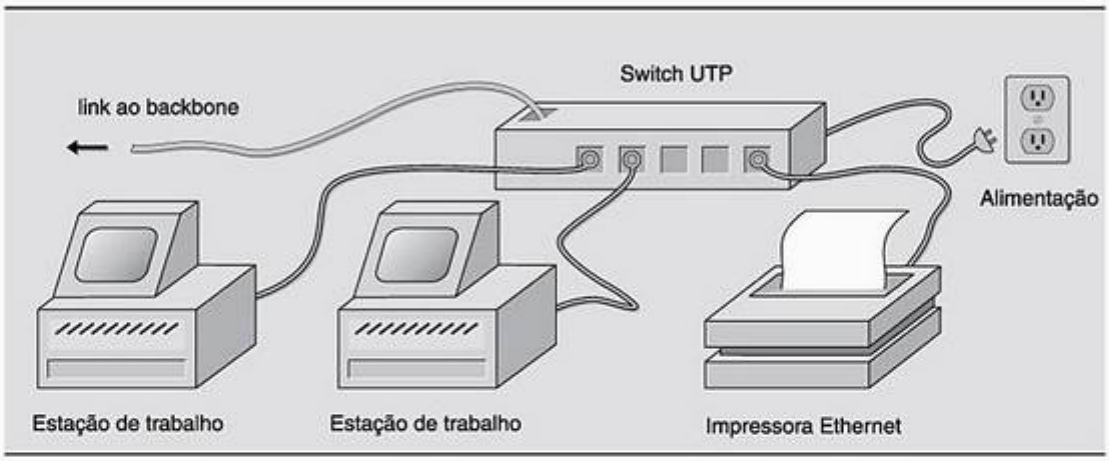
4. Não é brincadeira! Conectar um novo computador envolvia o processo de fazer um furo no revestimento externo do cabo com uma broca para se poder atingir o condutor central. Um "mordedor-vampiro" que morde o condutor externo era então fixado via parafusos.
5. Armazenamento temporário.

Snow666

- Ele usa fio de cobre barato e fácil de encontrar. (Algumas vezes, pode ser usada a fiação telefônica existente.)
- A fiação UTP é muito mais fácil de instalar e depurar do que fibra ou cabo coaxial. Comprimentos sob medida são obtidos facilmente.
- O UTP usa conectores RJ-45, que são baratos, confiáveis e fáceis de serem instalados.
- O link com cada máquina é independente (e privado!), de modo que um problema de cabeamento em um link dificilmente afetará outros hosts da rede.

O "formato" genérico de uma rede UTP é ilustrado no Diagrama B.

Diagrama B Uma instalação UTP



A fiação UTP adequada para uso em LANs modernas é comumente subdividida em oito classes. O sistema de avaliação de desempenho foi introduzido pela primeira vez pela Anixter, um grande fabricante de cabos. Esses padrões foram formalizados pela TIA (Telecommunications Industry Association, associação da indústria das telecomunicações) e são conhecidos hoje como Categorias 1 a 7, com uma Categoria 5E especial no meio.

A ISO (International Organization for Standardization, organização internacional de padrões) também entrou no excitante e rentável mundo da classificação de cabos e promove padrões exatamente equivalentes ou aproximadamente equivalentes às categorias TIA com numeração mais alta. Por exemplo, o cabo Categoria 5 da TIA é equivalente à Classe D da ISO. Para os *geeks* de plantão, a Tabela 15.2 ilustra as principais diferenças entre as diversas classificações modernas de hoje em dia. São informações interessantes de serem memorizadas para você poder impressionar seus amigos em festas.

Tabela 15.2 Características de cabos UTP.

Parâmetro	Categoria 5 Classe D[a]	Categoria 5E	Categoria 6 Classe E	Categoria 7 Classe F
Intervalo de freqüência	100 MHz	100 MHz	250 MHz	600 MHz
Atenuação	24 dB	24 dB	21.7 dB	20.8 dB
NEXT[b]	27.1 dB	30.1 dB	39.9 dB	62.1 dB
ACR[c]	3.1 dB	6.1 dB	18.2 dB	41.3 dB
ELFEXT[d]	17 dB	17.4 dB	23.2 dB	?[e]
Perda de retorno	8 dB	10 dB	12 dB	14.1 dB
Atraso de propagação	548 ns	548 ns	548 ns	504 ns

a. Inclui exigências TIA e ISO adicionais, TSB95 e FDAM 2, respectivamente.
b. Interferência próximo da extremidade.
c. Relação atenuação/interferência.
d. Interferência da ponta extrema de mesmo nível.
e. Este parâmetro atualmente não é especificado; estudo adicional pendente.

Snow666

Verifique a Seção 15.11 para obter maiores informações sobre cabeamento.

Na prática, os cabos Categoria 1 e 2 são adequados apenas para aplicações de voz (se tanto). Cabos Categoria 3 é o mínimo exigido para uma LAN; ele é o padrão para redes 10BaseT a 10 Mb/s. Cabos Categoria 4 é uma espécie de órfão, não adequado exatamente para qualquer tipo de aplicação em particular. Ele é usado ocasionalmente para redes Token Ring UTP a 16 Mb/s ou para instalações 10BaseT peculiares. Cabos Categoria 5 são capazes de suportar 100 Mb/s e é o padrão mais comumente usado hoje para cabeamento de dados. Os cabeamentos Categorias 5E e 6 suportam 1 Gb/s. Na época em que este livro era escrito, não havia nenhuma aplicação conhecida para os cabos Categoria 7, e nem todos os aspectos do padrão Categoria 7 (por exemplo, o formato físico dos conectores) haviam sido finalizados.

Conexões 10BaseT requerem dois pares de cabos Categoria 3, e cada link está limitado a um comprimento de 100 metros; conexões 100BaseTX possuem a mesma limitação de comprimento, mas requerem dois pares de cabos Categoria 5. Há cabos com revestimento de PVC ou de Teflon. Sua opção de revestimento deve se basear no ambiente no qual o cabo será instalado. Tubulações fechadas que passam pelo sistema de ventilação do prédio ("sistemas de retorno de ar") exigem, normalmente, Teflon.[6] O PVC é mais barato e mais fácil de trabalhar.

Conectores RJ-45 interligados com os pinos 1, 2, 3 e 6 são usados para estabelecer as conexões. Embora somente dois pares de fios sejam necessários para uma conexão funcional a 10 Mb/s ou 100 Mb/s, recomendamos que, ao instalar uma nova rede, você use fiação de quatro pares Categoria 5E e conecte todos os oito pinos do conector RJ-45.

Verifique a seção 7.1 para obter maiores informações sobre o padrão RS-232.

Para terminação de cabo UTP de quatro pares em painéis de interligação e tomadas de parede RJ-45, sugerimos que você use o padrão de fiação TIA/EIA-568A RJ-45. Esse padrão, que é compatível com outros empregos do RJ-45 (por exemplo, RS-232), é uma maneira conveniente de manter consistente a fiação em ambas as extremidades da conexão, independentemente de você poder ou não acessar facilmente os pares de cabos em si. O padrão 568A é detalhado na Tabela 15.3.

Tabela 15.3 Padrão TIA/EIA-568A para fiação UTP de quatro pares para um conector RJ-45.

Par	Cores	Conectados aos	Par	Cores	Conectados aos
1	Branco/Azul	Pinos 5/4	3	Branco/Verde	Pinos 1/2
2	Branco/Laranja	Pinos 3/6	4	Branco/Marrom	Pinos 7/8

Fiação já existente no prédio pode ou não ser adequada para emprego em redes, dependendo de como e quando ela foi instalada. Muitos prédios antigos foram reformados com cabos novos nos anos 50 e 60. Infelizmente, este tipo de cabo normalmente não oferecerá suporte nem mesmo a uma velocidade de 10 Mb/s.

Conectando e Expandindo redes Ethernet

As redes Ethernet podem estar conectadas logicamente em diversos pontos no modelo de rede OSI de sete camadas. Na camada 1, a camada física, podemos usar tanto conectores quanto repetidores físicos (comumente denominados hubs nos tempos modernos). Eles transferem o sinal diretamente, de forma muito parecida a duas latas conectadas por um barbante.

Na camada 2, a camada de link de dados, são usados switches. Os switches transferem frames tomando como base os endereços de origem e de destino, de forma muito semelhante a entregar uma mensagem numa garrafa, baseando-se somente no rótulo que se encontra por fora da garrafa.

Na camada 3, a camada de rede, são usados roteadores. Os roteadores transferem mensagens para o próximo salto de acordo com a localização do receptor final, semelhante a ler a mensagem dentro de uma garrafa para determinar a quem ela realmente é destinada.

Hubs e concentradores

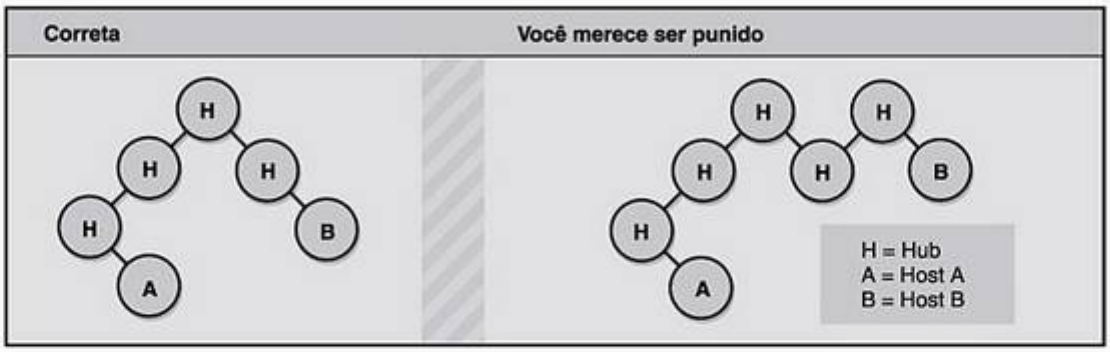
Hubs (também conhecidos como concentradores) são dispositivos ativos que conectam segmentos físicos em redes Ethernet UTP. Eles requerem alimentação de energia externa. Atuando como um repetidor, um hub ajusta novamente o tempo e reconstitui frames Ethernet, mas não os interpreta; ele não possui a mínima idéia para onde os pacotes estão indo ou que protocolo eles estão usando.

6. Fale com a brigada de incêndio de seu prédio para determinar as exigências de sua instalação.

Snow666

Os dois pontos mais distantes da rede jamais devem estar separados por mais de quatro hubs. O padrão Ethernet versões 1 e 2 especificavam no máximo dois hubs em série por rede. O padrão IEEE 802.3 estendeu esse limite para quatro em redes de 10 Mb/s. Redes Ethernet de 100 Mb/s permitem o emprego de dois hubs, redes Ethernet 1000BaseT somente um e redes 10GbE não permitem o emprego de hubs. O Diagrama C ilustra tanto uma configuração correta quanto uma incorreta para uma rede de 10 Mb/s.

Diagrama C Contagem de hubs



Os hubs requerem ocasionalmente atenção por parte de um administrador de sistemas, de modo que eles não devem ficar em lugares pouco iluminados e difíceis de serem alcançados. Reciclar a alimentação de energia (ou seja, desligando e ligando novamente) normalmente permite que eles se recuperem de um estado em que ocorreu travamento.

Switches

Os switches conectam redes Ethernet na camada de link de dados (camada 2) do modelo OSI. O propósito deles é o de conectar duas redes físicas diversas de uma maneira em que elas se pareçam com uma grande rede física. Eles não requerem software, mas sim recebem, regeneram e retransmitem pacotes em hardware.[7] A maioria dos switches usa um algoritmo de aprendizado dinâmico. Eles percebem quais endereços de origem provêm de uma determinada porta e quais provêm de uma outra. Os pacotes são encaminhados entre portas somente quando necessário. Inicialmente todos os pacotes são encaminhados, porém em poucos segundos o comutador já toma conhecimento das localizações da maioria dos hosts e poderá ser mais seletivo.

Já que nem todos os pacotes são encaminhados entre redes, cada segmento do cabo fica menos saturado com tráfego do que ficaria se todas as máquinas estivessem no mesmo cabo. Dada que a maior parte da comunicação tende a ser localizada, o aumento na largura de banda aparente pode ser dramático. E já que o modelo de rede não é afetado pela presença de um switch, há poucas conseqüências administrativas em instalar um.

Os switches podem, algumas vezes, ficar confusos caso a rede contenha loops. A confusão é originada pelo fato de os pacotes de um único host aparecerem em duas (ou mais) portas do switch. Uma rede Ethernet simples não pode conter loops, porém desde que conectemos várias redes Ethernet através de roteadores e switches, a topologia poderá incluir vários caminhos para um host. Alguns switches são capazes de lidar com essa situação reservando rotas alternativas no caso de a rota principal apresentar problemas. Eles executam uma operação de expurgo na rede até que as seções remanescentes apresentem somente um caminho para cada nó da rede. Alguns switches também são capazes de duplicar links entre as mesmas duas redes, e o tráfego de rotas é feito seguindo um modelo "round robin".

Os switches estão cada vez mais inteligentes à medida que mais recursos são embutidos em seus firmwares. Alguns podem ser usados para monitorar a segurança da rede. Eles registram qualquer endereço Ethernet externo que virem, conseqüentemente, detectando e relatando máquinas recém-conectadas. Já que operam na camada Ethernet, os switches são independentes de protocolo e podem lidar com qualquer combinação de tipos de pacotes de alto nível (por exemplo, IP, AppleTalk ou NetBEUI).

Os switches têm de varrer cada pacote para determinar se devem ser encaminhados. O desempenho dos switches é normalmente medido tanto pela taxa de varredura de pacotes quanto pela taxa de encaminhamento de pacotes. Muitos fornecedores não mencionam os tamanhos dos pacotes nos números que divulgam; conseqüentemente,

7. Pelo fato de os pacotes serem regenerados e recalculados em termos de tempo, redes totalmente comutadas não sofrem das limitações de "contagem de repetidores" exibidas no Diagrama C.

Snow666

o desempenho real pode ser menor do que aquele anunciado. Os switches se constituem numa maneira boa, porém ligeiramente cara, de se conectar redes Ethernet.

Embora o hardware de switches Ethernet esteja se tornando cada vez mais veloz, ele ainda não é uma tecnologia razoável para conexão de mais do que uma centena de hosts num único segmento lógico. Problemas como "tempestades de difusão ampla" normalmente afetam redes com muitos switches, já que o tráfego de difusão ampla tem de ser encaminhado a todas as portas num segmento com switch. Para solucionar esse problema, você deve usar um roteador para isolar o tráfego de difusão ampla entre segmentos com switches (criando, portanto, mais de uma rede Ethernet lógica).

Grandes instalações podem se beneficiar dos switches capazes de repartirem suas portas (através de configuração de software) em subgrupos chamados VLANs (Virtual Local Area Networks, redes locais virtuais). Uma VLAN é um grupo de portas que pertence ao mesmo segmento lógico, como se as portas estivessem conectadas aos seus próprios switches dedicados. Tal repartição aumenta a capacidade do switch isolar tráfego, e esta capacidade possui efeitos benéficos tanto em termos de segurança quanto de desempenho.

O tráfego entre VLANs é tratado por um roteador ou, em alguns casos, por um módulo de roteamento ou camada de software de roteamento dentro do switch. Uma extensão desse sistema conhecida como "entroncamento VLAN" (como aquele fornecido pelo protocolo IEEE 802.1Q) permite que switches separados fisicamente sirvam a portas na mesma VLAN lógica.

Escolher um switch pode ser difícil. O mercado de switches é um segmento altamente competitivo da indústria de informática e é infestado por anúncios de marketing que não são nem mesmo parcialmente verdadeiros. Ao selecionar uma marca de comutador, você deve confiar em avaliações independentes (como aquelas que aparecem em comparativos de revistas da área) em vez dos dados fornecidos pelos fabricantes em si. Em anos recentes, tem sido comum para um fabricante ter o "melhor" produto por alguns meses, e depois destruir completamente seu desempenho ou confiabilidade ao tentar realizar melhorias, fazendo com que um outro concorrente assuma a liderança.

De qualquer maneira, certifique-se de que a velocidade do backplane do switch esteja adequada — este é o número que efetivamente é considerado no final de um longo dia. Um switch bem projetado deve ter uma velocidade de backplane que exceda a soma da velocidades de todas as suas portas.

Roteadores

Os roteadores são computadores dedicados em um gabinete contendo duas ou mais interfaces de rede; eles dirigem tráfego na Camada 3 da pilha de protocolo OSI (a camada de rede). Eles enviam pacotes a seus destinos finais de acordo com as informações contidas nos cabeçalhos de protocolos TCP/IP. Além de simplesmente transferirem pacotes de um lugar para outro, os roteadores também podem executar outras funções como filtragem de pacotes (por questões de segurança), priorização (por questões de qualidade de serviço) e descoberta de topologia de rede como um todo. Veja os detalhes de como realmente funciona o roteamento no Capítulo 14.

Num único roteador podemos encontrar interfaces de hardware de vários tipos diferentes (por exemplo, FDDI, Ethernet e ATM). Em termos de software, alguns roteadores também são capazes de tratar de tráfego não-IP como IPX ou AppleTalk. Nessas configurações, o roteador e suas interfaces têm de ser configuradas para cada protocolo que você deseja que ele trate.

Os roteadores assumem uma de duas formas: configuração fixa e modular. Roteadores de configuração fixa possuem interfaces de rede específicas instaladas permanentemente de fábrica. Eles normalmente são adequados para aplicações pequenas e especializadas. Por exemplo, um roteador com uma interface T1 e uma interface Ethernet poderia ser uma ótima opção para conexão de uma pequena empresa à Internet.

Os roteadores modulares possuem uma arquitetura de barramento ou slots à qual as interfaces podem ser acrescentadas pelo usuário final. Embora essa forma normalmente seja cara, ela dá muito mais flexibilidade no futuro.

Dependendo de suas necessidades de confiabilidade e carga de tráfego esperada, um roteador dedicado pode ou não ser mais barato do que um sistema Linux configurado para atuar como um roteadore. Entretanto, o roteador dedicado normalmente fornecerá desempenho e confiabilidade superiores. Essa é uma área do projeto de redes na qual normalmente é aconselhável gastar um pouco mais no início para evitar dores de cabeça depois.

15.3 Wireless: A LAN do Nômade

As redes wireless formam uma área em rápido crescimento, e produtos em nível de produção agora estão se tornando disponíveis por preços vantajosos. Dado os recentes avanços na tecnologia de redes com conexão, as velocidades das redes wireless (indo normalmente de 2 Mb/s a 11 Mb/s) podem parecer um tanto inadequadas para o uso corporativo. Entretanto, essas velocidades são perfeitamente adequadas para vários fins. Uma rede wirelesss de 11 Mb/s numa residência ou pequeno escritório pode ser o sonho de qualquer administrador de sistemas. Além disso, o aces-

Snow666

so wireless para feiras, aeroportos e outros locais públicos pode realmente transformar um dia em que se fica incomunicável num dia hiperconectado para várias pessoas.

Os padrões mais promissores para redes wireless são o padrão para LANs wireless IEEE 802.11b (que fornecem acesso tipo LAN em velocidades de 2 Mb/s a 11 Mb/s a distâncias de 100 metros a 40 quilômetros, dependendo do equipamento e terreno) e o Bluetooth, um padrão para comunicação em distâncias curtas (usualmente menos do que 10 metros) e com baixa largura de banda (menos do que 1 Mb/s) entre PCs, periféricos, celulares e dispositivos portáteis como os PDAs.

Hoje, as redes 802.11b estão se tornando lugar comum. Os cartões são baratos e disponíveis para a maioria dos PCs laptop e desktop. Como acontece com as redes Ethernet com fio, a arquitetura mais comum para uma rede 802.11b usa um hub (chamado "ponto de acesso" no linguajar wireless) como ponto de conexão para uma série de clientes. Os pontos de acesso podem ser conectados a redes com fio tradicionais ou conectados de forma wireless a outros pontos de acesso (uma configuração conhecida como "malha wireless").

Você pode configurar um computador Linux para atuar como um ponto de acesso 802.11b caso tenha o hardware e o driver corretos (conhecemos pelo menos um chipset que oferece este tipo de suporte, o Intersil Prism II). Uma excelente estação-base wireless 802.11b dedicada para residências ou pequenos escritórios é a AirPort da Apple, um produto tipo nave espacial que é barato (em torno dos US$ 300) e altamente funcional. Compre uma hoje mesmo! Existem vários configuradores para o AirPort em sistemas diferentes do MacOS, entre os quais o Freebase (freebase.sourceforge.net) e um utilitário baseado em Java de Jonathan Sevy (edge.mcs.drexel.edu/ GICL/people/sevy/airport/).

A segurança de redes wireless hoje é muito pobre. O WEP (Wired Equivalent Privacy) é um protocolo usado em conjunto com redes 802.11b para fornecer tanto criptografia de 40 bits quanto de 128 bits para pacotes trafegando através de ondas aéreas. Infelizmente, a versão atual desse padrão demonstrou-se, há pouco tempo (agosto de 2001) ter uma falha grave de projeto (veja o site www.cs.rice.edu/~astubble/wep/wep_attack.html). Até que um padrão revisado se torne disponível, as redes wireless — com ou sem o WEP — devem ser consideradas completamente inseguras. Alguém próximo de seu prédio ou casa poderá acessar sua rede diretamente e de modo imperceptível.

Depurar uma rede wireless envolve também uma certa magia, já que uma ampla gama de variáveis estão em jogo quando da ocorrência de problemas. Em suma, as redes wireless são no momento interessantes e muito funcionais para residências, pequenos escritórios, conferências ou na praia, mas não irão substituir tão cedo os backbones corporativos que usam conexão física.

15.4 FDDI: A LAN DECEPCIONANTE E CARA

Em velocidades de 10 Mb/s, as redes Ethernet dos anos 80 não ofereciam largura de banda suficiente para algumas necessidades de rede, como conectar grupos de trabalhos por meio de um backbone corporativo (ou campus). Numa tentativa de fornecer opções com largura de banda maiores, o comitê ANSI X3T9.5 produziu a FDDI (Fiber Distributed Data Interface) como alternativa para redes Ethernet.[8] Desenvolvida e comercializada como um Token Ring de 100 Mb/s, a FDDI parecia ser uma solução fácil para as necessidades de largura de banda de muitas empresas.

Consulte a seção 13.3 para obter maiores informações sobre MTUs (Maximum Transmission Units).

Infelizmente, a FDDI tem sido uma decepção em todos os sentidos. Nos seus primórdios, o custo das interfaces FFDI normalmente excedia o das estações de trabalho em que elas estavam instaladas (por volta de US$ 10.000 cada) e o desempenho normalmente era pior do que o de interfaces Ethernet (placas iniciais da DEC, por exemplo). As interfaces ainda são caras, porém o desempenho está melhor. As interfaces modernas geram um desempenho por volta dos 80 Mb/s.

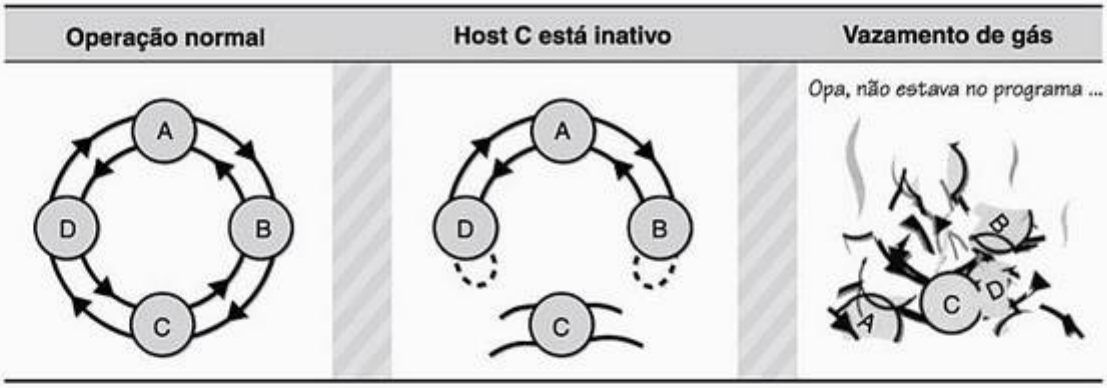
Para bom desempenho, a FDDI precisa de uma MTU muito maior do que o padrão que é ajustado para redes Ethernet. Um valor de MTU igual a 4.352 (configurado por meio de **ifconfig**) é razoável. Até que o software usado para transferir arquivos através de redes seja ajustado para a velocidade e para as características da FDDI, meros mortais provavelmente verão números de desempenho na faixa de um terço a metade do máximo teórico.

O padrão FDDI especifica uma LAN anel dual usando fibra ótica como meio de transmissão, com passagem de identificadores (tokens) a 100 Mb/s, conforme ilustrado no Diagrama D. A arquitetura de anel dual fornece um anel primário que é usado para transmissão de dados e um anel secundário que é usado como backup no caso de o anel ser cortado (tanto física como eletronicamente).

8. A FDDI também foi aceita como um padrão ISO.

Snow666

Diagrama D Token ring dual FDDI



Os hosts podem ser conectados a ambos os anéis (eles são chamados de hosts classe A ou hosts "conectados dualmente") ou simplesmente ao anel primário (hosts classe B ou hosts "conectados de forma simples"). Mais comumente, os concentradores e roteadores de backbone são conectados dualmente e as estações de trabalho conectadas de forma simples, em geral através de um "concentrador", uma espécie de hub para fibra ótica.

Uma vantagem dos sistemas token ring é que o acesso à rede é controlado por um protocolo determinístico. Não há colisões, de modo que o desempenho da rede não degrada sob carga intensa, como acontece com a Ethernet. Muitos sistemas token ring são capazes de operar de 90% a 95% de sua capacidade calculada ao servir a vários clientes.

Para mídia física, o padrão FDDI sugere dois tipos de fibra: modo simples e multimodo. Os "modos" são, basicamente, feixes de raios luminosos que entram na fibra num determinado ângulo. Fibras modo simples permitem que exatamente uma freqüência de luz trafegue pelo seu trajeto e, conseqüentemente, requer um laser como fonte emissora.[9] As fibras multimodo possibilitam vários trajetos e normalmente usam LEDs mais baratos e menos perigosos do que o laser. A fibra modo simples pode ser usada em distâncias maiores do que as fibras multimodo. Na prática, fibras multimodo 62,5 µm são mais comumente usadas para FDDI.

Vários padrões de conectores de fibra são usados com a FDDI e eles variam de fabricante para fabricante. Independentemente de quais conectores você use, tenha em mente que uma conexão de fibra adequada é essencial para operação confiável. Embora existam disponíveis "Kits faça você mesmo" para terminações de fibras, sugerimos que, sempre que possível, você empregue uma empresa especializada em cabeamento para instalar as terminações em trechos de fibra ótica.

15.5 ATM: A LAN PROMETIDA (PORÉM ALTAMENTE MALOGRADA)

ATM significa Asynchronous Transfer Mode (modo de transferência assíncrono), contudo alguns administradores insistem em deniominá-la Another Technical Mistake (mais um erro técnico). Um porta-voz dos fabricantes do mercado de comunicação de dados a descreveu como "uma tentativa da companhia telefônica de transformar o seu problema de rede em algo que eles sabem como tarifar".

A ATM é tecnicamente "especial", pois promove a filosofia de que pequenos pacotes de tamanho fixo (chamados "células") são a maneira mais eficiente de implementar redes na casa dos gigabits. A ATM também promete recursos que tradicionalmente não foram prometidos por outras mídias, inclusive garantias de qualidade de serviço e reserva de largura de banda.

A ATM foi amplamente divulgada como uma mídia de rede comutada tudo-em-um que poderia ser usada para requisitos de LAN, WAN e MAN. Nos tempos modernos, a ATM está praticamente morta, preservada somente em ambientes WAN em que grandes empresas de telecomunicações ainda estão tentando alavancar seus investimentos mal-orientados em hardware ATM.

Acima das células de 53 bytes da ATM, são descritas cinco AALs (ATM Adaptation Layers, camadas de adaptação ATM) para transporte de células. O propósito de cada camada de adaptação é sintetizado na Tabela 15.4 a seguir.

Não é claro como a AAL 2 alguma vez poderia ser usada na prática. Atualmente, não há nenhum padrão definido para ela. As AALs 3 e 4 acabaram ficando muito similares e foram combinadas. Um grupo de fabricantes que havia implementado a ATM estava insatisfeito com as AALs 3 e 4 devido ao alto custo adicional por elas imposto. Eles

9. *Jamais olhe diretamente nas extremidades de fibras cortadas ou pendentes*. Se elas forem a laser, poderão queimar seus olhos sem você perceber imeditamente.

Snow666

desenvolveram uma solução própria chamada SEAL (Simple and Efficient Adaptation Layer, camada de adaptação simples e eficiente) e que rapidamente se transformou na AAL 5.

Tabela 15.4 Camadas de adaptação ATM.

AAL	Aplicação
1	Aplicações com velocidade constante, como voz (requer atraso vinculado)
2	Aplicações em que a taxa de transmissão é variável requerendo atraso vinculado
3	Aplicações de dados orientadas a conexões
4	Aplicações de dados sem conexão
5	Transporte de dados genérico (especialmente tráfego IP, substitui 3 e 4)

15.6 Frame Relay: A WAN Sacrificatória

O frame relay é uma tecnologia WAN que oferece serviços de dados com comutação de pacotes, usualmente por um custo razoável. Embora a afirmação não seja 100% precisa, o frame relay normalmente é dito como um X.25 maquiado, uma assustadora tecnologia de comutação de pacotes da metade dos anos 70. Felizmente, ela é tão largamente usada que os equipamentos, software e pessoal que lhe dá suporte evoluíram para um estado de robustez e de bom desempenho.

Tradicionalmente, os usuários que desejavam conectar-se a sites remotos comprariam um circuito dedicado de uma companhia telefônica como, por exemplo, uma linha DDS de 56 Kb/s ou uma linha T1. Estes são circuitos de dados ponto-a-ponto que ficam conectados 24 horas por dia. Infelizmente, esse tipo de conexão normalmente é caro, já que requer que a companhia telefônica dedique equipamento e largura de banda ao link.

Por outro lado, o frame relay é uma abordagem de "economia de escala". A empresa de telefonia cria uma rede (normalmente conhecida como "nuvem"[10]) que estabelece a conexão com suas centrais. Usuários como você fornecem dados para sites remotos em pequenos pacotes. A companhia telefônica comuta os pacotes através das centrais apropriadas, em última instância entregando-os aos seus respectivos destinos. Nesse modelo, você e a empresa de telefonia estão apostando que, a cada segundo, a quantidade total de tráfego não excederá a largura de banda da rede (uma condição conhecida eufemisticamente como "estar subscrita em excesso").

Um roteador encapsula o tráfego IP através de conexões frame relay. Os pacotes são comutados através de PVCs (Permanent Virtual Circuits, circuitos virtuais permanentes) invisíveis, que permitem que os pacotes trafeguem somente aos sites para os quais você pagou para que eles chegassem. Esses PVCs fornecem algum grau de proteção de privacidade de outros sites conectados à rede frame relay.

A maior vantagem do frame relay é que normalmente ele é barato. Mas no mundo do "você recebe pelo que paga", você poderá descobrir que o desempenho de redes frame relay é, algumas vezes, sofrível. As conexões frame relay possuem custo adicional de comutação de pacotes, e a velocidade do link pode degradar em períodos de uso intenso.

15.7 ISDN: A WAN Nativa

O ISDN (Integrated Services Digital Network, rede digital de serviços integrados) é uma oferta das companhias telefônicas que assume diversas formas. Em sua forma mais comum e útil, chamada BRI (Basic Rate Interface), ISDN é essencialmente uma linha telefônica totalmente digital que fornece dois canais "B" discados de 64 Kb/s e um único canal "D" de 16 Kb/s para sinalização. Cada canal B pode ser usado tanto para dados como para voz (uma linha de voz pode ser transportada num único canal de 64 Kb/s).

O ISDN oferece uma linha digital de velocidade relativamente alta a um custo razoável (US$ 30–US$ 150 por mês, dependendo de onde você reside). Dispositivos chamados adaptadores de terminais convertem a linha telefônica numa interface mais familiar como a RS-232. Eles são usados (e cobrados) de maneira muito parecida com os modems. A maioria dos adaptadores pode agregar os dois canais B, levando um canal de dados de 128 Kb/s.

O ISDN pode ser usado no lugar de rede discada comum e também como tecnologia de rede WAN que utiliza um roteador ou bridge para conectar sites remotos através da linha.

10. Cujo nome é extremamente apropriado, já que jamais fica claro qual será a previsão meteorológica numa rede frame relay. Tempestades? Chuvas? Neve? Granizo?

Snow666

Embora muitas companhias telefônicas norte-americanas tenham instalado switches compatíveis com ISDN, elas ainda não descobriram como comercializá-los ou suportá-los.[11] Somente em poucas regiões podemos ligar para a companhia telefônica e solicitar uma linha ISDN. Algumas dicas: certifique-se de tratar com o departamento da companhia telefônica que cuida de serviços comerciais, já que é desta maneira como o ISDN é classificado no momento. Em muitas regiões você terá de vencer várias ondas de parasitas antes de conseguir falar com alguém que realmente conhece o ISDN, mesmo que o serviço realmente não esteja disponível.

15.8 DSL e Modems a Cabo: A WAN das Pessoas

É fácil transferir grandes quantidades de dados entre empresas e outras grandes instalações de dados. Tecnologias oferecidas por empresas de telefonia como T1, T3, SONET, ATM e frame relay fornecem canais relativamente simples para transferência de dados de um lugar a outro. Entretanto, estas tecnologias não são opções realistas para conectar residências e escritórios domésticos. Elas custam muito e a infra-estrutura necessária não se encontra largamente disponível.

O DSL (Digital Subscriber Line, linha de assinante digital) usa cabos telefônicos de cobre comuns para transmissão de dados em velocidades de até 7 Mb/s (embora conexões DSL típicas levem a velocidades dentre 256 Kb/s e 768 Kb/s). Uma vez que a maioria das residências já possui cabos telefônicos instalados, o DSL é uma maneira de fornecer a "última milha" de conectividade da companhia telefônica até a residência. As conexões DSL normalmente terminam numa "caixa" que atua como um roteador TCP/IP e fornece conexão Ethernet para outros dispositivos dentro da residência. O DSL normalmente é mais barato e também mais rápido do que o ISDN, de modo que se constitui hoje na tecnologia preferida para usuários domésticos.

Diferentemente de conexões POTS (Plain Old Telephone Service) comuns e conexões ISDN, que requerem que você "disque" para um destino final, o DSL é um serviço de rede dedicado que está sempre conectado. Esse recurso o torna ainda mais atraente, pois não há nenhuma configuração ou atraso de conexão quando um usuário quer transferir dados.

O DSL vem em várias formas e, conseqüentemente, ele com freqüência é chamado de xDSL, com o x representando uma subtecnologia específica como A de assimétrico, S de simétrico, H de alta velocidade ou RA de taxa de transferência adaptável. As variantes de tecnologia e taxas de transferência de dados exatas disponíveis em sua região dependem dos equipamentos que as empresas de telefonia optaram por empregar.

A corrida pela conectividade da "última milha" para centenas de milhões de residências é uma corrida disputada. Ela é também altamente politizada, bem capitalizada e superdivulgada. A abordagem DSL alavanca a infra-estrutura de cobre que é comum entre os ILECs (Incumbent Local Exchange Carriers), que preferiram altas margens de lucro em vez de investimentos em infra-estrutura conforme a revolução das redes dos anos 80 e 90 passou por elas.

As empresas de TV a cabo, que já possuem infra-estrutura de fibra ótica na maioria dos bairros, estão promovendo suas próprias soluções de "última milha", que levam a conexões de banda larga similares (embora assimétricas) para as residências. A indústria de modems a cabo recentemente se tornou esclarecida sobre padrões de dados e está promovendo no momento o padrão DOCSIS (Data Over Cable Service Interface Specification). Esse padrão define as especificações técnicas tanto para modems a cabo quanto para os equipamentos usados na empresa de TV a cabo e ele permite que várias marcas de equipamentos operem entre si.

Em suma, a briga entre as tecnologias de modem a cabo e DSL em grande parte se resume a "meu orçamento para marketing é maior do que o seu". O DSL possui uma certa vantagem inerente, já que cada conexão é privada para um determinado cliente; os modems a cabo num bairro dividem largura de banda e podem ser capazes de fazer escuta clandestina no tráfego alheio.

15.9 Para Onde as Redes Estão Caminhando?

Ao observar de perto as tecnologias descritas até aqui, você observará uma coisa em comum: as mais simples e mais baratas estão sendo bem-sucedidas, ao passo que as mais complexas e mais caras estão morrendo rapidamente. Onde isso nos coloca no futuro?

O padrão Ethernet deu uma surra em seus rivais, pois ele é incrivelmente barato. Ele é tão simples de ser implementado que hoje até podemos comprar fornos de microonda com interfaces Ethernet. O padrão Ethernet cresceu bem: em muitas empresas, a infra-estrutura Ethernet de 10 Mb/s do início dos anos 80 ainda está em uso produtivamente, conectada a trechos de 100 Mb/s e de 1 Gb/s. A Ethernet de 10 Gb/s já saiu dos laboratórios e logo estará disponível em larga escala. Esperamos ver essa tendência prosseguindo, com hardware de comutação cada vez mais rápido para conectar tudo isso.

---

11. Daí a seguinte interpretação: ISDN = It Still Does Nothing, ou seja, ele ainda não faz nada.

www.snow666.blogspot.com
Snow666

No front da "conectividade para o lar", o DSL dá novo ânimo às velhas e cansadas instalações de cabos de cobre. A proliferação de modems a cabo trouxe acesso de alta velocidade (e sérios problemas de segurança) ao alcance de milhões de residências.

O que é fantástico em relação a todos esses novos desenvolvimentos é que independentemente da mídia ou de sua velocidade, o TCP/IP é compatível com eles.

15.10 Teste e Depuração de Redes

A principal vantagem da migração em larga escala para a Ethernet (e outras tecnologias baseadas em UTP) é a facilidade de depuração das redes. Já que essas redes podem ser analisadas link por link, problemas de rede normalmente podem ser isolados em segundos em vez de dias.

O segredo para depurar uma rede é subdividi-la em suas partes componentes e testar cada parte até você ter isolado o dispositivo ou cabo problemático. As "luzes idiotas" dos switches e hubs (como as de "estado de link" e as de "tráfego de pacotes") normalmente nos dão dicas imediatas para chegar à fonte do problema. Documentação de primeira categoria sobre seus esquemas de fiação é essencial para fazer com que essas luzes indicadoras trabalhem a seu favor.

Como acontece com a maioria das tarefas, ter as ferramentas corretas para o serviço é meio caminho andado para ter o trabalho feito de forma correta e sem atrasos. O mercado oferece dois tipos principais de ferramentas para depuração de problemas em redes (embora elas estejam crescendo juntas rapidamente).

A primeira delas é o analisador de cabos portátil. Esse dispositivo é capaz de medir as características elétricas de um dado cabo, inclusive seu comprimento (com uma tecnologia maravilhosa chamada TDR (time domain reflectrometry, refletometria de domínios de tempo). Normalmente, esses analisadores também podem apontar falhas simples como cabos rompidos ou mal conectados. Nosso produto favorito para análise de cabos de LANs é o Fluke LanMeter. Trata-se de um analisador tudo-em-um capaz de realizar até mesmo pings IP através da rede. Versões topo de linha possuem seu próprio servidor Web capaz de exibir históricos estatísticos. Para circuitos WAN (telco), o analisador de linha T-Berd é o "miado" do gato. O T-Berd e seu correspondente topo de linha para testes de LANs, o FireBERD, são produzidos pela TTC (www.ttc.com). Esteja ciente de que todos esses produtos custam tanto quanto uma pequena casa na maioria das cidades.

O segundo tipo de ferramenta é o analisador (sniffer) de redes. Esse dispositivo desmonta pacotes de rede em busca de erros de protocolo, configurações incorretas e desordens em geral. Normalmente é exigido um nível substancial de treinamento e paciência para usar um dispositivo desses de maneira efetiva, e nos dias de hoje análise em nível de pacotes de rede raramente é necessária. Caso queira fazê-la, o Cadillac dos analisadores de redes é fabricado pela Sniffer Technologies (www.sniffer.com).

15.11 Fiação Predial

Independentemente de você estar em uma rede Ethernet na casa dos gigabits ou apenas com cabos seriais, recomendamos que use cabos de melhor qualidade possível. Isso aumentará as chances de você ainda usar a mesma fiação por dez anos. É mais barato executar-se a fiação no prédio inteiro de uma só vez em vez de fazê-lo conexão por conexão.

Opções de cabeamento UTP

Cabos Categoria 5E são relativamente novos e oferecem a melhor relação custo/benefício no mercado atual. Seu formato normal é o de quatro pares por revestimento, que é apropriado para uma série de conexões de dados, indo desde RS-232 até Ethernet na casa dos gigabits.

As especificações da Categoria 5E requerem que o trançado seja mantido dentro do intervalo de meia polegada da conexão ao bloco punchdown. Isso implica que qualquer fiação com mais de quatro pares por revestimento terá de ser amarrada com fita ou presa para manter o trançado, já que alimenta mais do que uma conexão.

Você tem de usar terminadores Categoria 5E além dos cabos Categoria 5E. Tivemos a maior sorte em usar peças fabricadas pela The Siemon Company of Watertown, Connecticut (www.siemon.com).

Conexões para escritórios

Uma conexão por escritório certamente não é o suficiente. Mas você deve usar duas ou quatro? Recomendamos quatro, por diversas razões:

- Elas podem ser usadas para conexões seriais (modem, impressora etc.).
- Elas podem ser usadas com telefones.
- Elas podem ser usadas para acomodar visitantes ou máquinas de demonstração.

Snow666

- O custo dos materiais gira em torno de somente 5% a 10% do custo total.
- Sua melhor estimativa multiplicada por dois é uma boa estimativa.
- É muito mais barato fazer isto de uma só vez do que ficar acrescentando fiação depois.

Se estiver no processo de realizar a fiação de todo o seu prédio, talvez seja interessante considerar a instalação de algumas tomadas nos corredores, salas de reunião, refeitórios e banheiros. As redes estão invadindo todos os cantos.

Padrões de fiação

Os prédios modernos normalmente requerem uma infra-estrutura de fiação grande e complexa para suportar todas as diversas atividades que ocorrem internamente. Entrar num gabinete de telecomunicações mediano normalmente é uma experiência chocante para alguém de estômago fraco, já que fios sem identificação e da mesma cor normalmente cobrem as paredes.

Numa tentativa de aumentar a rastreabilidade e de padronizar a fiação predial, a TIA (Telecommunications Industry Association) lançou o TIA/EIA-606 Administration Standard para infra-estrutura de telecomunicações em prédios comerciais em fevereiro de 1993. O padrão EIA-606 especifica as exigências e as diretrizes para a identificação e documentação de infra-estrutura de telecomunicações.

Entre os itens abrangidos pelo EIA-606 temos:

- Hardware para terminações
- Cabos
- Caminhos de cabos
- Espaços para equipamentos
- Codificação de cores para infra-estrutura
- Símbolos para componentes-padrão

Em particular, ele especifica cores-padrão a serem usadas para fiação. Os detalhes ocultos são revelados a seguir na Tabela 15.5.

A Pantone agora vende software para fazer o mapeamento entre sistemas Pantone para papel que recebe tinta, matizes têxteis e plásticos coloridos. Ei, você poderia coordenar as cores da fiação, os uniformes dos instaladores e a documentação sobre fiação! Pensando melhor…

15.12 Questões de Projeto

Esta seção trata do projeto lógico e físico de uma rede. Ela é destinada a instalações de porte médio. As idéias aqui apresentadas serão indicadas para algumas centenas de hosts, mas são um exagero para três máquinas e insuficientes para milhares de máquinas.

Tabela 15.5 Tabela de cores EIA-606.

Tipo de terminação	Cor	Código[a]	Comentários
Ponto de demarcação	Laranja	150C	Terminações do escritório central
Conexões de rede	Verde	353C	Também usado para terminações de circuitos auxiliares
Equipamento comum[b]	Púrpura	264C	Principais terminações para equipamentos de dados/comutação
Backbone de primeiro nível	Branco	–	Terminações de cabos
Backbone de segundo nível	Cinza	422C	Terminações de cabos
Estação	Azul	291C	Terminações de cabos horizontais
Backbone entre prédios	Marrom	465C	Terminações de cabos campus
Variado	Amarelo	101C	Manutenção, alarmes etc.
Sistemas telefônicos de teclas	Vermelho	184C	–

a. De acordo com o Pantone Matching System®
b. PBXes, hosts, LANs, muxes etc.

Snow666

Também supomos que você tenha um orçamento suficiente e esteja começando da estaca zero, o que provavelmente seja apenas parcialmente verdadeiro.

A maior parte de um projeto de rede consiste na especificação do seguinte:

- Tipos de mídia que serão utilizados
- A topologia e trajeto percorrido pelos cabos
- O emprego de repetidores, bridges e roteadores

Uma outra questão fundamental no projeto de redes é o controle de congestionamento. Por exemplo, o NFS impõe uma carga bem pesada à rede e, portanto, servir arquivos em um cabo backbone é indesejável.

As questões apresentadas nas seções a seguir são tipicamente aquelas que devem ser consideradas no projeto de qualquer rede.

Arquitetura de rede *versus* Arquitetura do prédio

A arquitetura de rede normalmente é mais flexível do que a arquitetura predial, porém as duas têm de coexistir. Se você tiver sorte suficiente para poder especificar a rede antes de o prédio ser construído, seja pródigo. Para a maioria de nós, tanto o prédio quanto o departamento de gerenciamento de instalações já existem e são um tanto rígidos.

Em prédios existentes, a rede tem de usar a arquitetura do prédio, e não brigar contra ela. Os prédios modernos normalmente contêm dutos para passagem de cabos de dados e telefônicos, além daqueles da fiação elétrica de alta voltagem e encanamentos de água e gás. Normalmente eles usam forros, uma bênção para instaladores de rede. Muitas empresas possuem dutos subterrâneos que facilitam a instalação de redes.

A integridade das paredes corta-fogo[12] deve ser preservada; se você passar um cabo através de uma parede/porta corta-fogo, o furo tem de ser justo e preenchido com uma substância não combustível. Respeite os sistemas de retorno de ar ao escolher o seu cabo. Caso você seja pego violando legislação de incêndio, poderá ser multado e será exigido que você corrija os problemas criados, mesmo que isto signifique inutilizar a rede inteira e refazê-la corretamente.

O projeto lógico de sua rede tem de estar de acordo com as restrições físicas impostas pelas edificações que ela atende. Ao especificar uma rede, tenha em mente que é fácil desenhar uma solução logicamente boa e depois descobrir que é fisicamente difícil senão impossível de implementá-la.

Redes existentes

As redes de computadores são o foco desta discussão, porém muitas organizações já possuem redes CATV e redes telefônicas capazes de transmitir dados. Normalmente, estas incluem links por fibra ótica. Se sua organização estiver pronta para instalar um sistema telefônico novo, compre grandes quantidades de fibra a mais e as instale ao mesmo tempo.

Tivemos a oportunidade, há vários anos, de perguntar aos empreiteiros se eles passariam alguns cabos de fibra ótica para nós. Eles disseram: "Claro, sem cobrar nada extra" e ficaram um tanto zangados ao mostrarmos a eles a tonelada de cabos de fibra ótica que eles teriam de instalar.

Expansão

É muito difícil prever necessidades daqui a dez anos, especialmente nos campos da computação e de suas redes. Portanto, é importante projetar a rede tendo em mente os fatores expansão e aumento na largura de banda. À medida que o cabo for sendo instalado, especialmente em lugares fora do caminho e de difícil acesso, puxe de três a quatro vezes o número de pares que você realmente precisa no momento. Lembre-se: a maior parte do custo de instalação reside na mão-de-obra, não nos materiais. Mesmo que você não tenha planos imediatos para usar fibra ótica, é sagaz instalar uma certa extensão ao fazer a fiação de seu prédio, especialmente se for difícil de instalar cabos mais tarde. Puxe tanto fibra modo simples quanto multimodo; certamente o tipo que você precisará no futuro sempre será aquele que você não instalou.

Congestionamento

Uma rede é como uma corrente: somente tão boa quanto o seu elo mais fraco ou mais lento. O desempenho de redes Ethernet, como muitas outras arquiteturas de rede, degrada à medida que a rede é carregada.

---

12. Esse tipo de compartimento é feito de concreto, tijolo ou material que retarda as chamas impedindo-as de se propagarem e queimarem o prédio. Embora um tanto diferente de um firewall para segurança de rede, provavelmente ele tem a mesma importância.

Snow666

Switches sobrecarregados, interfaces cujos tipos são discordantes e links de baixa velocidade são fatores que levam ao congestionamento de uma rede. Vale a pena isolar tráfego local criando sub-redes e usando dispositivos de interconexão como roteadores. As sub-redes também podem ser usadas para isolar máquinas que são usadas para experimentação; é difícil executar um experimento que envolva várias máquinas se não houver uma maneira fácil de isolar essas máquinas tanto física quanto logicamente do restante da rede.

Manutenção e documentação

Descobrimos que a manutenção de uma rede está altamente correlacionada com a qualidade de sua documentação. Documentação acurada, completa e atualizada é absolutamente indispensável.

Os cabos devem ser identificados em todos os pontos de terminação e também em intervalos regulares de alguns metros para poderem ser facilmente identificados quando forem descobertos num forro ou parede.[13] É uma boa idéia deixar cópias dos diagramas de cabeamento locais dentro dos gabinetes de comunicações para que estes diagramas possam ser atualizados no local quando as alterações forem feitas. Depois, em intervalos regulares de algumas semanas, alguém deve copiar essas alterações para entrada num banco de dados eletrônico.

As junções entre os principais centros populacionais na forma de switches ou roteadores podem facilitar a depuração possibilitando que partes da rede sejam isoladas e depuradas separadamente. Também é útil colocar junções entre domínios políticos e administrativos.

15.13 Questões de Gerenciamento

Se a rede deve funcionar corretamente, algumas coisas precisam ser centralizadas, outras distribuídas e outras ainda devem ser locais. Há necessidade de formular e de haver acordo sobre regras concretas razoáveis e de "política de boa vizinhança".

Um ambiente típico incluiria o seguinte:

- Uma rede backbone entre os prédios
- Sub-redes departamentais conectadas ao backbone
- Sub-redes de grupos dentro de um departamento
- Conexões para o mundo externo (por exemplo, Internet ou escritórios externos)

Várias facetas do projeto e da implementação de redes têm de ter controle, responsabilidade, manutenção e financiamento que abranja toda a instalação. As redes com algoritmos de tarifação segmentada para cada conexão crescem de modo muito bizarro, porém muito previsível à medida que os departamentos tentam minimizar seus próprios custos locais. Os principais alvos para controle centralizado são:

- O projeto da rede, incluindo o emprego de sub-redes, roteadores, switches etc.
- O cabo backbone em si, inclusive as conexões para ele
- Endereços IP de hosts, nomes de host e nomes de subdomínios
- Protocolos, basicamente para garantir interoperabilidade
- Política de roteamento para a Internet

Os nomes de domínios, endereços IP e nomes de rede são, a certo ponto, já controlados por autoridades como ARIN e ICANN. Porém, o uso desses itens por parte de sua instalação também deve ser coordenado localmente.

Uma autoridade central possui uma visão global da rede: seu projeto, capacidade e crescimento esperado. Ela é capaz de arcar com equipamentos próprios de monitoramento (e o pessoal para comandá-los) e de manter a saúde da rede backbone. Ela pode insistir em projeto de rede correto, mesmo quando isto significar dizer a um departamento para comprar um roteador e criar uma sub-rede para conexão com a rede principal. Tal decisão poderia ser necessária de maneira que uma nova conexão não impactasse negativamente a rede existente.

Se uma rede atende a vários tipos de máquinas, sistemas operacionais e protocolos, é quase essencial ter um roteador muito inteligente (por exemplo, Cisco) como gateway entre redes.

15.14 Fornecedores Recomendados

Ao longo de mais de 15 anos instalando redes ao redor do mundo, acabamos nos dando mal algumas vezes por usarmos produtos que não atendiam às especificações ou foram mal representados, eram muito caros ou de alguma forma deixaram de atender às nossas expectativas. Apresentamos a seguir uma lista de fornecedores como referência,

13. Alguns fabricantes de cabos já identificam previamente rolos de cabos em intervalos de alguns metros.

Snow666

que ainda confiamos, recomendamos e nós próprios usamos hoje em dia. Apesar de estarem sediados nos Estados Unidos, a maioria deles possui escritórios em diversos países.

Cabos e conectores

AMP
P.O. Box 3608
Harrisburg, PA 17105
www.amp.com

Anixter
4711 Golf Rd.
Skokie, IL 60076
(708) 677-2600
www.anixter.com

Belden Cable
P.O. Box 1980
Richmond, IN 47375
www.belden.com

Krone
6950 S. Tucson Way
Englewood, CO 80112
www.krone.com

Black Box Corporation
P.O. Box 12800
Pittsburgh, PA 15241
www.blackbox.com

Newark Electronics
4801 N. Ravenswood Ave.
Chicago, IL 60640
(312) 784-5100
www.newark.com

The Siemon Company
76 Westbury Park Road
Watertown, CT 06795
www.siemon.com

Equipamento de testes

Wavetek
9045 Balboa Ave.
San Diego, CA 92123
www.wavetek.com

Fluke
P.O. Box 9090
Everett, WA 98206
www.fluke.com

The Siemon Company
76 Westbury Park Road
Watertown, CT 06795
www.siemon.com

TTC
20400 Observation Drive
Germantown, Maryland 20876
www.ttc.com

Roteadores/Switches

Cisco Systems
PO Box 3075
1525 O'Brien Drive
Menlo Park, CA 94026-1435
www.cisco.com

15.15 Leitura Recomendada

GROTH, DAVID; JIM MCBEE. *Cabling: The Complete Guide to Network Wiring*. Sybex. 2000.

SEIFERT, RICH. *Gigabit Ethernet*. Reading, MA: Addison-Wesley. 1998.

ANSI/TIA/EIA-568-A, *Commercial Building Telecommunications Cabling Standard*, e ANSI/TIA/EIA-606, *Administration Standard for the Telecommunications Infrastructure of Commercial Buildings*, são os padrões da indústria de telecomunicação para a fiação em edifícios. Infelizmente, eles não são gratuitos. Veja www.tiaonline.org.

SPURGEON, CHARLES. "Guide to Ethernet."
http://wwwhost.ots.utexas.edu/ethernet/ethernet-home.html

Snow666

15.16 Exercícios

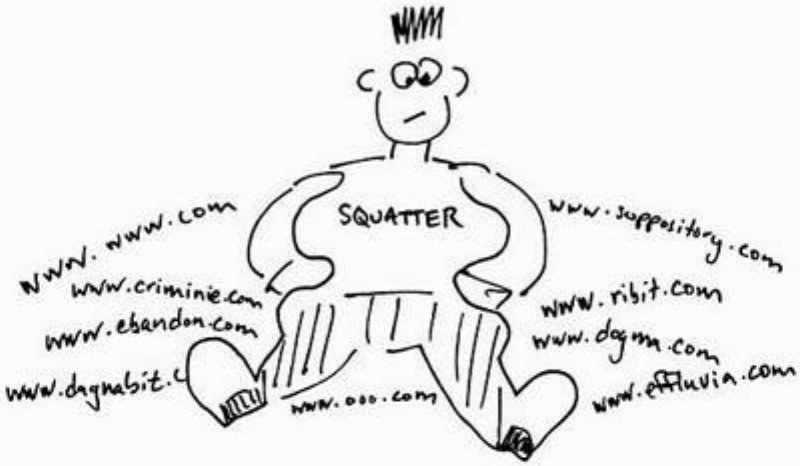
E15.1 Hoje, a maioria dos edifícios para escritórios abriga redes de computadores e estas são conectadas via Ethernet UTP. Algumas combinações de hubs e switches são necessárias para suportar essas redes. Em muitos casos, os dois tipos de equipamentos são intercambiáveis. Liste as vantagens e as desvantagens de cada um deles.

★ **E15.2** Desenhe um diagrama de redes simples e imaginário que conecte uma máquina em sua rede experimental à Amazon.com. Inclua componentes LAN, MAN e WAN. Mostre que tecnologia é usada para cada um desses componentes. Ilustre alguns hubs, switches e roteadores.

★★ **E15.3** TTCP é uma ferramenta que mede o desempenho de redes TCP e UDP (procure-a em www.rpmfind.net ou com apt-cache no Debian). Instale o TTCP em duas máquinas ligadas em rede e meça o desempenho do link entre elas. O que acontece à largura de banda se você aumentar ou diminuir os tamanhos dos buffers? Como os números obtidos se comparam em relação à capacidade teórica da mídia física?

Snow666

16 O Sistema de Nomes de Domínios



Zilhões de hosts estão conectados à Internet. Como controlar todos eles quando pertencem a tantos países, redes e grupos administrativos diferentes? Duas peças importantes da infra-estrutura juntam todas as coisas: o DNS (Domain Name System, sistema de nomes de domínios), que controla quem são os hosts, e o sistema de roteamento Internet, que controla como eles estão conectados.

Este capítulo trata da porção DNS deste sistema. Embora o DNS tenha surgido para atender a diversos propósitos, sua tarefa principal é a de associar nomes de hosts a endereços IP. Usuários e programas em nível de usuário adoram se referir a máquinas pelo nome, porém o software de rede de baixo nível entende somente números. O DNS fornece o elo que faz com que todo o mundo fique contente. Ele também passa a desempenhar um papel importante no direcionamento de e-mail e acesso a servidores Web.

O DNS é um banco de dados distribuído. "Distribuído" significa que meu site armazena os dados sobre meus computadores, o seu site armazena dados sobre os seus computadores e, de alguma forma, nossos sites cooperam e compartilham dados automaticamente entre si quando um site precisa procurar dados do outro.

16.1 DNS para o Impaciente: Acrescentando uma Nova Máquina

Este capítulo é praticamente um minilivro por si só. Antes de mergulharmos em suas profundezas capazes de entorpecer nossas mentes, vamos dar uma respirada para responder à pergunta mais comum sobre DNS: Como acrescentar um novo host a uma rede que já está usando DNS? A receita a seguir lhe mostra como fazê-lo copiando e modificando os registros existentes para um computador similar — templatehost.my.domain.

Etapa 1: Escolha um nome de host e endereço IP para a nova máquina em conjunto com administradores de sistemas locais ou com o seu ISP (Internet Service Provider, provedor de serviços Internet).

Etapa 2: Identifique uma máquina similar na mesma sub-rede. Usaremos os registros dessa máquina como modelo para nossas máquinas novas.

Etapa 3: Faça o login para a máquina mestre de servidor de nomes.

Etapa 4: Pesquise em **/etc/named.conf** ou **/etc/bind/named.conf** (Debian):

- Por meio da instrução `options`, localize a linha `directory` que informa onde arquivos de dados de zona são mantidos em seu site (veja a seção 16.9). Os arquivos de zona contêm o host em si e os dados de endereço IP.
- Por meio das instruções `zone`, localize os nomes de arquivos para o arquivo de zona direta (forward) e o arquivo de zona reversa da rede em que o seu novo endereço IP se encontra (seção 16.9).

Snow666

Etapa 5: Vá para o diretório de arquivos de zonas e edite o arquivo de zona direta (usando RCS e **sudo**, é claro). Localize os registros para o host-modelo que você identificou antes. Eles serão parecidos com o seguinte:

```
templatehost    IN    A  128.138.243.100
                IN    MX 10 mail-hub
                IN    MX 20 templatehost
```

Etapa 6: Duplique esses registros e altere-os de maneira apropriada para o seu novo host. O arquivo de zonas pode estar ordenado segundo o nome de host; siga a convenção existente. Certifique-se também de alterar o número de série no registro SOA no início do arquivo (ele é o primeiro dos cinco números no registro SOA). O número de série somente deve aumentar; adicione 1 caso seu site use um número de série arbitrário ou configure o campo para a data atual caso seu site use esta convenção.

Etapa 7: Edite o arquivo de zonas reversas,[1] duplique o registro para o host-modelo e atualize-o. Ele deve ser parecido com o seguinte:

```
100             IN    PTR   templatehost.my.domain.
```

Você também tem de atualizar o número de série no registro SOA do arquivo de zonas reversas.

Se o seu arquivo de zonas reversas apresentar mais do que o último byte do endereço IP de cada host, você terá de introduzir os bytes na ordem inversa. Por exemplo, o registro

```
100.243         IN    PTR   templatehost.my.domain.
```

corresponde ao endereço IP 128.138.243.100 (neste caso, a zona reversa é relativa a 138.128.in-addr.arpa, e não a 243.138.128.in-addr.arpa).

Etapa 8: Enquanto ainda estiver logado na máquina mestre de servidor de nomes, execute **ndc reload** (**rndc reload** para BIND 9).

Etapa 9: Verifique a configuração usando **dig**; veja a seção 16.14. Você também poderá tentar **ping** ou **traceroute** para o nome de seu novo host, mesmo que o novo host ainda não tenha sido configurado. Uma mensagem "host unknown" (máquina desconhecida) indica que você errou; "host not responding" (máquina não respondendo) significa que provavelmente tudo está OK.

Os erros mais comuns são:

- Esquecer de atualizar o número de série e reiniciar o servidor de nomes; e
- Esquecer de acrescentar um ponto no final do nome de host na entrada reversa PTR.

16.2 A História do DNS

Nos bons tempos, a associação (mapeamento) entre nomes de hosts e endereços era mantida num único arquivo de texto controlado de forma centralizada e distribuído a todos os hosts da ARPANET. Os nomes de host não eram hierárquicos e o procedimento para atribuir um nome a um computador incluía verificar se ninguém mais no mundo havia usado o nome que você desejava usar. As atualizações consumiam grande parte da largura de banda da ARPANET e o arquivo estava constantemente desatualizado.

Logo ficou claro que embora uma tabela de hosts estática fosse razoável para uma rede pequena, ela era inadequada para a grande ARPANET em constante crescimento. O DNS soluciona os problemas de uma tabela estática usando dois conceitos básicos: nomes de host hierárquicos e responsabilidade distribuída. O DNS foi inicialmente especificado por Paul Mockapetris nas RFCs 882 e 883 (1983) e atualizado nas RFCs 1034 e 1035 (1987). Paul também criou uma implementação inicial para sistemas não-UNIX.

O trabalho UNIX original foi feito por quatro universitários da Berkeley (Douglas Terry, Mark Painter, David Riggle e Songnian Zhou) em 1984. Ele foi então escolhido por Ralph Campbell do Computer Systems Research Group de Berkley, que iniciou sua colagem no BSD. Em 1985, Kevin Dunlap, um engenheiro da DEC temporariamente cedido à Berkeley, assumiu o projeto e produziu o sistema BIND (Berkeley Internet Name Domain). Mike Karels, Phil Almquist e Paul Vixie fizeram a manutenção do BIND ao longo dos anos. Ele vem com a maioria dos sistemas UNIX ou Linux e também pode ser encontrado em www.isc.org.

O ISC (Internet Software Consortium) é uma organização sem fins lucrativos que mantém diversas peças cruciais de software Internet, inclusive o BIND. Paul Vixie e Mark Andrews mantêm atualmente a árvore de código

1. A zona reversa poderia ser mantida em qualquer outro lugar (por exemplo, no site de seu provedor de acesso). Se isso acontecer, a entrada reversa terá de ser introduzida lá.

Snow666

BIND 8 em nome do ISC com a ajuda de participantes da lista de discussão de trabalhadores do BIND. O ISC desenvolveu o BIND 9 financiado por vários fabricantes, órgãos governamentais e outras organizações.

O ISC também fornece vários tipos de apoio a esses produtos, inclusive ajuda na configuração, classes para BIND e DNS e até mesmo programação personalizada. Esses serviços são uma bênção para sites que são obrigados a terem um contrato de suporte antes de poderem usar o código aberto. Várias empresas usam contratos de serviço como uma maneira de contribuírem para o ISC — eles adquirem contratos caros, mas jamais pedem ajuda.

As RFCs 1034 e 1035 ainda são consideradas a especificação-base para o DNS, porém mais de 30 outras RFCs as suplantaram e acrescentaram diversos aspectos ao protocolo e registros de dados ao longo da última década (veja a lista no final deste capítulo). Atualmente, nenhum padrão ou RFC única consegue juntar todos os pedaços num único local. Historicamente, o DNS foi mais ou menos definido como "o que o BIND implementar", de maneira que este vem se tornando menos preciso à medida que surgem outros servidores DNS.

Embora o DNS tenha sido implementado em sistemas operacionais diferentes do UNIX/Linux, este livro discute somente o BIND. A Nortel portou o BIND para Windows e o retribuiu ao ISC; desde então, o BIND versão 8.2 ficou disponível também para Windows. Graças à padronização do protocolo DNS, implementações DNS para UNIX e para sistemas não-UNIX são capazes de operarem entre si e de compartilharem dados. Muitos sites rodam servidores UNIX para fornecerem serviço DNS a seus desktops Windows; a combinação funciona bem.

A Microsoft fornece um servidor DNS no Windows 2000, porém a implementação da Microsoft possui suas peculiaridades e diferenças próprias. Ele interage com o BIND, mas também tende a entulhar a rede com pacotes desnecessários ou malformados.

16.3 Quem Precisa do DNS?

O DNS define:

- Um espaço de nomes hierárquico para hosts e endereços IP
- Uma tabela de hosts implementada na forma de um banco de dados distribuído
- Um "solucionador" – rotinas de biblioteca que consultam este banco de dados
- Encaminhamento de e-mails otimizado
- Um mecanismo para localizar serviços em uma rede
- Um protocolo para intercâmbio de informações de atribuição de nomes

Para serem verdadeiros cidadãos da Internet, os sites precisam do DNS. Manter um arquivo **/etc/hosts** local com mapeamentos para cada host que você imagina querer manter contato é impossível.

Cada site mantém um ou mais blocos do banco de dados distribuído que forma o sistema DNS mundial. O seu bloco de banco de dados é formado por dois ou mais arquivos de texto contendo registros para cada um de seus hosts. Cada registro é uma linha única formada por um nome (normalmente um nome de host), um tipo de registro e alguns valores de dados. O campo de nome pode ser omitido caso seu valor seja o mesmo da linha anterior.

Por exemplo, as linhas a seguir

```
forklift        IN    A     192.108.21.7
                IN    MX    10 chimchim.xor.com.
```

no arquivo "direto" (forward) e

```
7               IN    PTR   forklift.xor.com.
```

no arquivo "reverso" associam "forklift.xor.com" ao endereço IP 192.108.21.7 e redirecionam e-mails endereçados a essa máquina para o host chimchim.xor.com.

O DNS é um sistema cliente/servidor. Servidores ("servidores de nomes") carregam os dados de seus arquivos DNS para a memória e os usam para responder a consultas tanto de clientes internos como de clientes externos e de outros servidores na Internet. Todos os seus hosts devem ser clientes DNS, porém relativamente um pequeno número deles precisa ser servidor DNS.

Se sua organização for pequena (alguns hosts numa única rede), você poderá rodar um servidor em um host ou solicitar a seu ISP para fornecer serviço DNS em seu nome. Use um servidor autorizado e recursivo caso você mesmo forneça serviço DNS, e um servidor somente para cache (ou nenhum servidor local em si) caso seu ISP forneça este tipo de serviço. Um site de porte médio com várias sub-redes deve manter vários servidores DNS para reduzir a latência de consultas e aumentar a confiabilidade. Um site muito grande poderá dividir seu domínio DNS em subdomínios e manter vários servidores para cada subdomínio.

Snow666

16.4 NOVIDADES NO DNS

Várias mudanças significativas foram feitas no DNS ao longo dos últimos anos. Esta seção lhe dá uma amostra das principais modificações e um guia para os locais onde elas são cobertas com mais detalhes.

Tanto o DNS quanto o BIND estão sendo constantemente atualizados. O DNS possui novos tipos de registros de recursos, novos ajustes finos para protocolos e algumas características novas. O BIND foi reprojetado e reescrito com suporte para multithreads e sistemas multiprocessados. A Tabela 16.1 lista as principais alterações.

Tabela 16.1 Novos recursos no DNS e BIND.

Página	RFCs	Recurso
421	2052	Registros SRV para localização de serviços
423	–	Registros A6 para endereços IPv6
424	2672–3	Registros DNAME para redirecionamento de pesquisa de endereços IPv6
418	2317	Delegação in-addr sem classes (o hack CNAME)
424[a]	–	O domínio ip6.arpa para mapeamentos IPv6 reversos
-[a]	–	Um solucionador que reconhece IPv6
376	2671	EDNS0, mudanças e extensões de protocolo
384	1996	Notificação assíncrona de mudanças de zonas
433	2136	Atualização dinâmica (para sites que usam DHCP)
432	1995	Transferências de zonas incrementais
440	2535–41	DNSSEC, autenticação e segurança para dados de zonas
438	2845	Assinaturas de transações TSIG/TKEY e intercâmbio de chaves

a. Não é visto neste livro ou, no caso do ip6.arpa, não é abordado com muitos detalhes.

Algumas dessas novas características são projetos enormes que o IETF ainda não finalizou sua padronização. Os grupos de trabalho que estão criando os padrões possuem bons escritores, porém carecem de batalhadores vigilantes em termos de codificação; algumas das especificações mais recentes podem ser difíceis ou até mesmo impossíveis de serem implementadas. A versão atual do BIND (8.2.5) inclui alguns dos recursos novos; o release inicial do BIND 9 (9.2.0) inclui quase todos eles, contudo, não necessariamente em suas formas finais.

O IPv6 é descrito de forma mais detalhada no Capítulo 13.

Dois novos recursos tremendos, suporte ao IPv6 e DNSSEC, justificam algum comentário. O IPv6 aumenta o comprimento dos endereços IP de 32 bits para 128 bits. Se alguma vez ele for totalmente implementado, terá um impacto enorme sobre a Internet. O BIND 9 suporta os trechos do IPv6 que já foram padronizados até então, porém parece pouco provável que o IPv6 será usado amplamente durante o período de vida deste livro. Portanto, nossa cobertura sobre o suporte a IPV6 do BIND 9 será breve. Há bastante neste capítulo para que você sinta o gostinho, mas não o suficiente para lhe garantir uma migração de seu site para o IPv6 e para configuração do DNS para ele.

O padrão DNSSEC tenta acrescentar autenticação ao banco de dados DNS e seus servidores. Ele usa criptografia de chave pública para verificar a origem e a integridade dos dados DNS e usa o DNS para distribuir chaves, bem como dados de host.

Também foram introduzidos mecanismos de autenticação mais simples, tais como suporte para autenticação através do emprego de um "código secreto compartilhado". Contudo, o código secreto compartilhado tem de ser distribuído a cada par de servidores que quiser executar autenticação mútua. Embora isso seja aceitável para um site local com um punhado de servidores, o mesmo não é verdade em nível de Internet. O BIND 9 implementa tanto sistema de chaves públicas DNSSEC quanto o sistema de código secreto compartilhado TSIG (Transaction SIGnatures, assinaturas transacionais).

Esperamos ver alguma forma de autenticação usada amplamente nos próximos anos, começando com as zonas raiz. Experimentos têm demonstrado que é possível obter-se assinaturas em poucas horas em algumas zonas de primeiro nível (por exemplo, nl e de). Entretanto, a zona **com** é tão grande que levaria dias para se obter assinaturas com a tecnologia atual. Já que agora **com** é atualizada duas vezes ao dia, um processo de assinatura que leva dias não funcionaria. Ataques de segurança contra a integridade do DNS sem dúvida nenhuma acelerarão a adoção de medidas de autenticação.

No horizonte para o DNS está a introdução de nomes de domínios internacionalizados, que permitirá que países em que a língua não é o inglês definam nomes de domínios em seus alfabetos nativos. Cada uma destas três gran-

Snow666

des questões (IPv6, DNSSEC e internacionalização) aumenta significativamente o tamanho dos registros de dados DNS e, portanto, tornando mais provável que o DNS acabará chegando aos limites de tamanho dos pacotes UDP.

16.5 O Espaço de Nomes DNS

Nas próximas seções, discutiremos a anatomia geral do DNS (a especificação) e depois descreveremos os arquivos de configuração usados pelo BIND (a implementação). Caso você já esteja familiarizado com o DNS e queira ir diretamente ao assunto do capítulo, vá para a seção 16.8 *Questões Relativas a Clientes no BIND* ou para a seção 16.9 *Configuração de Servidores BIND*.

No mundo real e em qualquer ponto deste livro, veremos os termos DNS e BIND sendo usados de forma intercambiável. Porém, neste capítulo tentamos (talvez sem sucesso) preservar a distinção existente entre eles.

O espaço de nomes DNS é uma árvore de "domínios". Cada domínio representa um trecho diferente do espaço de nomes e é controlado de maneira livre por uma única entidade administrativa. A raiz da árvore é chamada de "." ou ponto, e abaixo dele se encontram os domínios de primeiro nível (ou domínios em nível-raiz). Os domínios de primeiro nível foram relativamente fixos no passado, porém o ICANN aprovou sete novos domínios de primeiro nível em 2001: biz, info, name, pro, museum, aero e coop.[2] Na época da publicação deste livro, museum estava operacional, pro disponível somente para registro e os demais em processo no DNS para se tornarem operacionais.

Um ramo da árvore de nomes associa nomes de hosts a endereços IP e um segundo ramo associa endereços IP de volta a seus nomes de host. O primeiro ramo é chamado de "mapeamento direto" e os arquivos de dados BIND associados a ele são chamados de "arquivos de zonas diretas". O ramo endereço-nome de host é chamado de "mapeamento reverso" e seus arquivos de dados são chamados de "arquivos de zonas reversas".

Por razões históricas, dois tipos de nomes de domínio de primeiro nível estão sendo usados hoje em dia. Nos Estados Unidos, os domínios de primeiro nível originalmente descreviam entidades organizacionais e políticas e recebiam nomes de três letras como com e edu. Alguns desses domínios (principalmente com, org e net) também são usados fora dos Estados Unidos; eles são chamados de domínios genéricos de primeiro nível ou gTLDs, de forma abreviada.

A Tabela 16.2 lista os gTLDs mais importantes juntamente com seus objetivos originais. Quando bons nomes no domínio com se tornaram escassos, os órgãos de registro começaram a oferecer nomes nos domínios org e net sem levar em conta as restrições originais destes domínios. Os domínios da coluna esquerda da Tabela 16.2 são os originais, datando de cerca de 1988; a coluna da direita inclui os novos domínios acrescentados em 2001.

Tabela 16.2 Domínios genéricos de primeiro nível.

Domínio	A que se destina	Domínio	A que se destina
com	Empresas comerciais	aero	Segmento de transporte aéreo
edu	Instituições educacionais	biz	Negócios
gov	Órgãos governamentais	coop	Cooperativas
mil	Órgãos militares norte-americanos	info	Uso irrestrito
net	Provedores de rede	museum	Museus
org	Organizações sem fins lucrativos	name	Indivíduos
int	Organizações internacionais	pro	Contadores, advogados etc.
arpa	Árvore de endereços IP		

Para a maioria dos domínios fora dos Estados Unidos, são usados os códigos ISO de duas letras para países; eles são chamados de ccTLDs. Os TLDs geográficos e organizacionais coexistem dentro do mesmo espaço de nomes global. A Tabela 16.3 ilustra alguns códigos de países comuns.

Alguns países fora dos Estados Unidos criam uma hierarquia organizacional com domínios de segundo nível. As convenções para atribuição de nomes variam. Por exemplo, uma instituição acadêmica poderia ser um edu nos Estados Unidos e um ac.jp no Japão.

2. ICANN é a sigla de Internet Corporation for Assigned Names and Numbers, o órgão regulador da Internet. Veja a seção 13.1 para maiores informações sobre o ICANN.

www.snow666.blogspot.com
Snow666

Tabela 16.3 Códigos de países comuns.

Código	País	Código	País	Código	País
au	Austrália	fi	Finlândia	hk	Hong Kong
ca	Canadá	fr	França	ch	Suíça
br	Brasil	jp	Japão	mx	México
de	Alemanha	se	Suécia	hu	Hungria

O domínio de primeiro nível "us" também é usado algumas vezes nos Estados Unidos, basicamente com domínios de localidade; por exemplo, bvsd.k12.co.us, a Boulder Valley School District no Estado de Colorado. O domínio "us" jamais é combinado com um domínio educativo — não há nenhum "edu.us" (ainda). A vantagem dos nomes de domínios "us" se baseia no fato de serem gratuitos ou baratos para se fazer o registro; veja www.nic.us para obter maiores detalhes. As restrições em domínios de segundo nível abaixo de "us" (que antigamente eram limitadas aos Estados dos EUA) estão sendo relaxadas, e nomes de domínios como evi-nemeth.us em breve serão possíveis.

Os mercenários de domínios em alguns casos compram o espaço de nomes de um país inteiro. Por exemplo, o domínio para a Moldávia, "md", agora está sendo comercializado para médicos e residentes do Estado de Maryland (MD) nos Estados Unidos. Um outro exemplo é Tuvalu, para o qual o código de país é "tv". A primeira de tais vendas foi Tonga ("to"), a mais ativa no momento é Niue ("nu") e, quem sabe, a mais atraente seja "tm" de Turquemenistão. Essas negociatas algumas vezes foram razoáveis para o país com o código de duas letras desejado e outras vezes não.

A grilagem de domínios é largamente praticada: o pessoal registra nomes que eles acham que serão solicitados no futuro e depois os revendem para as empresas cujos nomes afanaram. Anos atrás, todas as estações de esqui do Colorado foram registradas em nome do mesmo indivíduo, que fez um bom dinheiro revendendo-os para várias estações de esqui individuais na época em que estes últimos quiseram ter seus sites na Web. O preço médio para um bom nome de domínio com está na faixa de vários milhares de dólares a alguns poucos milhões de dólares — business.com foi vendido por US$ 3,5 milhões.

Ofereceram-nos US$ 50.000 para o nome admin.com, que conseguimos anos atrás quando sysadmin.com já havia sido adquirido pela revista /Sys/Admin.

Os nomes de domínios não diferenciam as letras maiúsculas de minúsculas. "Colorado" é o mesmo que "colorado", que é o mesmo que "COLORADO" no que se refere a DNS. As implementações DNS atuais ignoram a diferenciação entre letras maiúsculas e minúsculas, mas a propagam quando esta é fornecida.[3] No passado era comum usar letras maiúsculas para os domínios de primeiro nível e a primeira letra em maiúsculo para domínios de segundo nível. Nos dias de hoje, os dedos já estão cansados de digitar e a norma é tudo em letra minúscula.

Dois recursos novos do DNS entram em choque em relação à distinção entre maiúsculas e minúsculas: a internacionalização de nomes e a segurança DNSSEC. Nomes internacionalizados requerem que o tipo da letra (maiúscula ou minúscula) tenha significado e seja preservado, porém o DNSSEC transforma todos os nomes para letras minúsculas antes de computar assinaturas criptográficas. É provável que o DNS irá canonizar internamente nomes para letras minúsculas para seus cálculos criptográficos, mas enviará os dados em si com o tipo dos caracteres (maiúsculo ou minúsculo) preservado. Qualquer codificação internacional terá de incluir regras de canonização. Com sorte, o pessoal de padronização do IETF solucionará essas questões antes de qualquer um desses novos recursos estar em uso em larga escala.

O nome de um host Internet totalmente qualificado é formado anexando-se seu nome de domínio a seu nome de host. Por exemplo, boulder.colorado.edu é o nome totalmente qualificado para o host boulder na Universidade de Colorado. Outros sites podem usar o nome de host boulder sem entrar em conflito, pois os nomes totalmente qualificados serão diferentes.

Dentro do sistema DNS, os nomes totalmente qualificados terminam com um ponto como, por exemplo, "boulder.colorado.edu.". A falta de um ponto final indica um endereço relativo. Dependendo do contexto no qual um endereço relativo é usado, componentes extras poderiam ser adicionados. De fato, alguns sistemas (como correio eletrônico) travarão se você fornecer o ponto por conta própria. É comum para um host ter mais de um nome. O host boulder.colorado.edu também poderia ser conhecido como www.colorado.edu ou ftp.colorado.edu caso quiséssemos fazer com que seu nome refletisse os serviços por ele fornecidos. Na realidade, é uma boa prática tornar "móveis" nomes de host de serviços como www, de modo que você poderá deslocar servidores de uma máquina para

3. BIND (a implementação de referência usada pela maioria dos sites) preserva a distinção da letra (maiúscula ou minúscula), porém algumas implementações (por exemplo, o DNS da Microsoft e DJB) mudam de tipo de letra (maiúscula ou minúscula) de acordo com suas próprias preferências. Isso é o bastante para o assunto padrões rigorosos.

www.snow666.blogspot.com
Snow666

outra sem alterar o nome primário de qualquer máquina. A atribuição de nomes extras é obtida pela construção CNAME; verifique seção16.11.

Quando recebemos o nome colorado.edu, nos foi garantido que colorado era exclusivo dentro do domínio edu. Nós dividimos ainda mais esse domínio em subdomínios de acordo com os departamentos. Por exemplo, o host anchor no Departamento de Ciências da Computação chama-se anchor.cs.colorado.edu na Internet.

A criação de cada subdomínio tem de ser coordenada com os administradores do domínio acima para garantir exclusividade. As entradas nos arquivos de configuração para o domínio-pai delegam autoridade para o espaço de nomes ao subdomínio.

Mestres de seus domínios

O gerenciamento de domínios de primeiro nível com, org, net e edu, nos Estados Unidos, era antigamente coordenado pela Network Solutions, Inc., sob contrato da National Science Foundation. Essa situação de monopólio agora mudou e outras organizações têm permissão para registrarem nomes de domínios nesses gTLDs. Outros domínios de primeiro nível, como aqueles para cada país, são mantidos por organizações regionais.

Houve várias propostas para permitir que empresas privadas operassem seus próprios domínios de primeiro nível, e é provável que domínios de primeiro nível adicionais estarão disponíveis num futuro próximo. Consulte www.icann.org para obter informações atualizadas.

A maioria dos ISPs oferece serviços de registro de nomes de domínios tarifados. Eles lidam com a autoridade de domínios de primeiro nível em seu nome e configuram seus servidores DNS para tratarem das pesquisas de nomes dentro de seu domínio. Embora você possa reduzir despesas diretas tratando diretamente com os registros e rodando os seus próprios servidores DNS, você não estará necessariamente poupando dinheiro. A desvantagem de depender dos servidores de um ISP é o fato de você perder o controle direto sobre a administração de seu domínio.

Mesmo que você queira gerenciar os seus próprios serviços DNS, ainda terá de se adequar ao seu provedor de acesso. A maioria dos ISPs fornece mapeamentos DNS reversos para endereços IP dentro de seus blocos CIDR. Se você assumir o gerenciamento de seus endereços DNS, certifique-se de que seu ISP desabilite o seu serviço para estes endereços e delegue esta responsabilidade a você.

Veja a seção 13.4 para obter maiores informações sobre o CIDR.

Os mapeamentos direto e reverso de um domínio devem ser gerenciados no mesmo local sempre que possível. Alguns ISPs ficam contentes em deixar sob sua responsabilidade o gerenciamento dos arquivos direto, porém são relutantes em abandonarem o controle de mapeamentos reversos. Esse gerenciamento dividido pode levar a problemas de sincronização. Veja a seção 16.11 para uma alternativa elegante que faz com que a delegação funcione até mesmo para pequenos trechos de espaço de endereços.

Os domínios DNS devem (são obrigados, na realidade; veja a RFC1219) ser servidos por pelo menos dois servidores. Um arranjo comum é aquele em que um site opere o seu próprio servidor mestre e deixe que os servidores do ISP atuem como backup. Assim que o sistema tiver sido configurado, os servidores do ISP farão o download automático das informações de configuração por meio de seu servidor mestre. Alterações feitas à configuração DNS se refletem nos servidores de backup sem qualquer intervenção explícita por parte dos administradores de ambos os sites.

Não coloque todos os seus servidores DNS na mesma rede. Quando o DNS pára de operar, a rede, efetivamente, pára aos olhos de seus usuários. Distribua seus servidores DNS de modo que você não acabe ficando com um sistema frágil com um único ponto de falha. O DNS é bem robusto se configurado de maneira cuidadosa.

Selecionando um nome de domínio

Certos nomes são um tabu; por exemplo, nomes que já foram usados. Outros que costumavam estar fora dos limites recentemente receberam permissão, como combinações de domínios de primeiro nível (edu.com) e domínios com componentes repetidos (x.x.com[4] ou com.com[5]).

Nosso conselho costumava ser que os nomes deveriam ser os mais curtos e fáceis de serem digitados e que deveriam identificar a organização que os usava. Nos dias de hoje, a realidade é que todos os nomes bons e curtos já fo-

4. Nem todos os nomes com componentes repetidos eram inválidos. Por exemplo, xinet.xinet.com sempre foi um nome válido. O trecho referente ao domínio é xinet.com e o domínio contém um host chamado xinet.
5. Num certo ponto, os servidores para com.com eram todos considerados defeituosos (veja a seção *Delegações defeituosas*). Infelizmente, os sistemas Windows têm o costume de acrescentarem .com a nomes que eles não conseguem encontrar. Uma configuração comum resultou em computadores NT consultando servidores gTLD 5.000 vezes por segundo em busca de nomes no domínio com.com. Trata-se de uma carga excessiva de consultas não solicitadas pelos usuários, isto é, fabricadas desnecessariamente pelo NT.

Snow666

ram usados por alguém, pelo menos no domínio com. É tentador jogar a culpa nos grileiros, mas na realidade a maioria dos bons nomes está efetivamente em uso.

A RFC1032 recomenda que os nomes de domínios de segundo nível não sejam maiores do que 12 caracteres, porém o DNS na verdade permite até 63 caracteres em cada componente e até 255 caracteres num nome completo. A sugestão de 12 caracteres é normalmente ignorada e realmente não há nenhuma razão para aderir a ela a não ser o de aliviar o tédio de digitação de nomes longos.

O inchamento dos domínios

O DNS foi desenvolvido para associar o nome de domínio de uma organização a um servidor de nomes para esta organização. Desse modo ele precisa ser capaz de ser ampliado ao número de organizações existentes no mundo. Porém, agora que a Internet se tornou um conduíte de cultura de massa, os nomes de domínios estão sendo aplicados a todos os produtos, filmes, eventos esportivos, substantivos em inglês etc. Nomes de domínios como twinkies.com não estão (diretamente) relacionados à empresa que fabrica o produto; eles simplesmente estão sendo usados como propagandas. Não está claro que o DNS seja capaz de continuar a crescer dessa forma exponencial. O problema real aqui é que a árvore de atribuição de nomes DNS é uma estrutura de dados eficiente somente quando ela tem alguma forma de hierarquia e não seja totalmente plana. Com cada empresa atribuindo centenas ou até mesmo milhares de produtos no nível superior da árvore, a hierarquia está condenada.

O que realmente precisamos é de um serviço de diretórios que associe marcas e nomes-fantasia a empresas, deixando o DNS livre para lidar com os endereços IP. Os princípios dessa idéia são implementados nos navegadores mais modernos através de um serviço fornecido pela RealNames Corporation. Infelizmente, a RealNames é um monopólio proprietário; somente empresas que fazem uma assinatura paga poderão ter suas palavras-chave listadas no banco de dados. Uma outra solução possível é forçar a hierarquia no sistema; por exemplo, twinkies.hostess-foods.com. Porém isto jamais ocorrerá — já fomos muito longe na trilha marketing-domínio-nome.com.

A Sony faz as coisas certas segundo a perspectiva do DNS — todos os seus produtos são subdomínios de sony.com. Pode significar um ou dois cliques a mais para localizar os produtos desejados, mas o DNS aprecia a hierarquia.

Registrando um nome de domínio de segundo nível

Para obter um nome de segundo nível, você tem de solicitar à autoridade Internet apropriada do domínio de primeiro nível. O ICANN está credenciando atualmente varias agências para fazerem parte de seu projeto de escritórios de registro compartilhado para se registrar nomes nos gTLDs. Na época em que este livro era escrito, tínhamos algo em torno de 200 opções de agências de registro. Vá ao endereço www.icann.org para obter uma lista definitiva.

Na Europa, contacte o Council of European National Top-level Domain Registries no endereço www.centr.org para identificar o escritório de registro local e reivindicar um nome de domínio. Para a região Ásia-Pacífico, o organismo apropriado é o Asia-Pacific Network Information Center, www.apnic.net. No Brasil, o organismo apropriado é o Registro.br, www.registro.br

Para completar os formulários de registro de domínios, você tem de identificar uma pessoa para contatos técnicos, uma outra para contatos administrativos e pelo menos dois hosts que serão os servidores para o seu domínio. Você também terá de escolher um nome que ainda não tenha sido escolhido.

Criando seus próprios subdomínios

O procedimento para criação de um subdomínio é similar àquele para criação de um domínio de segundo nível, exceto que a autoridade central agora é local (ou mais precisamente, dentro de sua própria organização). Especificamente, as etapas são as seguintes.

- Escolha um nome que seja exclusivo no contexto local.
- Identifique dois ou mais hosts para serem servidores de seu novo domínio.
- Coordene-se com o administrador do domínio-pai.

Domínios-pai devem fazer uma verificação para garantir que os servidores de nomes de um domínio-filho estejam ativos e funcionais antes de fazer a delegação. Se os servidores não estiverem funcionando, resultará numa "delegação defeituosa" e pode ser que você receba um e-mail desagradável solicitando que corrija o seu procedimento DNS. A seção 16.14 analisa as delegações defeituosas com mais detalhes.

www.snow666.blogspot.com
Snow666

16.6 O SOFTWARE BIND

BIND (Berkeley Internet Name Domain) é um pacote de software de código aberto da ISC que implementa o protocolo DNS e fornece serviço de nomes em sistemas Linux, UNIX, MacOS e Windows.

Versões do BIND

Houve três versões principais do BIND: BIND 4, BIND 8 e BIND 9. O BIND 4 surgiu no final dos anos 80 (correspondendo grosseiramente ao lançamento das RFCs 1034 e 1035). O BIND 8 foi lançado em 1997 e o BIND 9 em meados de 2000. Não há nenhum BIND 5, 6 ou 7; o BIND 8 foi uma atualização tão significativa que os autores sentiram que ela merecia um número de versão que fosse o dobro da anterior.[6] Bem, não foi realmente assim ... o BIND 8 foi lançado com o 4.4BSD, para o qual todos os números de versão foram elevados para 8. **sendmail** também pulou alguns números e foi para a versão 8 na mesma época.

O BIND 8 incorporou numerosos avanços técnicos que aumentaram a eficiência, robustez e segurança. O BIND 9 aumenta ainda mais a aposta com suporte a multiprocessamento, operação segura com threads, segurança efetiva (criptografia de chaves públicas), suporte ao IPv6, transferências incrementais de zonas e uma enormidade de outros recursos. Uma nova estrutura de dados (pelo menos nova para o BIND), a árvore vermelha-preta, armazena dados de zonas na memória. O BIND 9 é uma reimplementação e reprojeto completos. Ele isola as partes do código específicas ao sistema operacional, tornando mais fácil portar o BIND para sistemas não-UNIX. As partes internas do BIND 9 são significativamente diferentes, porém seu procedimento de configuração permanece o mesmo.

O BIND 4 é mantido somente em relação às correções de segurança e em breve será descontinuado. Espera-se que após um ou dois anos depois do BIND 9 tornar-se estável e de uso comum, o BIND 8 também seja suspenso. Ficamos tentados a discorrer somente sobre o BIND 9 neste livro, porém, já que a linguagem de configuração da V9 é um superconjunto da V8 e já que também não temos tanta experiência prática com a V9, tratamos das duas versões.

Muitos sites postergaram a atualização por estarem hesitantes em acabar prejudicando programas que estavam funcionando bem. Caso você ainda esteja usando o BIND 4, o script Perl **named-bootconf.pl** que vem com as distribuições V8 e V9 pode converter um arquivo de configuração V4 em seu equivalente V8 ou V9.

O verdadeiro banco de dados de registros DNS não precisa ser alterado. Um arquivo de configuração convertido da versão 4 não usará nenhum dos novos recursos do BIND 8 e do BIND 9, mas pode ser um bom ponto de partida de onde você expandir.

Descobrindo que versão você possui

Normalmente não parece ocorrer aos fabricantes documentar que versão de um pacote de software externo eles incluíram em seus sistemas, de maneira que você pode ter que fazer uma investigação para descobrir exatamente com que versão de software está trabalhando. No caso do BIND, podemos algumas vezes determinar o número de versão com uma consulta furtiva usando **dig**, um comando que vem com o BIND. O comando **dig @***servidor* **version.bind txt chaos** retorna o número de versão a menos que alguém tenha decidido restringir esta informação alterando-a no arquivo de configuração. Por exemplo, o comando funciona em isc.org:

```
  dig @rc.isc.org version.bind txt chaos
version.bind.      0S CHAOS TXT    "9.2.0rc5"
```

mas não em cs.colorado.edu:

```
  dig @mroe.cs.colorado.edu version.bind txt chaos
version.bind.      0S CHAOS TXT "wouldn't you like to know…"
```

Alguns sites configuram o BIND para esconder seu número de versão baseado na teoria que isto fornece algum grau de "segurança através de obscuridade". Realmente não endossamos essa prática, mas isso poderia ajudá-lo a rechaçar alguns dos scripts maliciosos. Veja a seção 16.9 para obter uma discussão mais detalhada sobre esse tópico.

Consulte o Capítulo 11 para obter maiores informações sobre syslog.

Também normalmente podemos descobrir que versão temos instalada inspecionando os arquivos de log em **/var/log**. O daemon de servidor BIND, **named**, registra em log seu número de versão para syslog (recurso "daemon") quando ele é iniciado. Faça um **grep** para linhas como a seguinte (este é do Red Hat):

```
Nov 5 17:17:58 coyote named[8076]: starting (/etc/named.conf). named 8.2.3-
   REL Sat Jan 27 05:53:47 EST 2001 root@porky.devel.redhat.com:
   /usr/src/bs/BUILD/bind-8.2.3/src/bin/named
```

6. Quem disse que os departamentos de Marketing e de Engenharia não podem conviver pacificamente?

Snow666

Se **named** estiver instalado, mas seu sistema não se iniciar normalmente no momento do boot, basta executá-lo manualmente como root sem nenhum argumento. **named** registrará em log seu número de versão, descobrirá que não possui nenhum arquivo de configuração e encerrará.

A Tabela 16.4 mostra as versões do BIND incluídas nas distribuições usadas como exemplo neste livro. Versões anteriores a 8.2.3 são conhecidas por apresentar problemas de segurança.

Tabela 16.4 Versões do BIND nos sistemas usados como exemplo neste livro.

Sistema	Versão do Sistema Operacional	Versão do BIND
ISC	–	8.2.5/9.2.0
Red Hat	7.2	9.1.0-10
SuSE	7.3	8.2.4-REL
Debian	3.0	8.3.0-REL-NOESW[7]

O Red Hat é conhecido por congelar números de versão e depois instalar correções colocando-os, portanto, fora de sincronismo com o restante da comunidade. Os números de versão internos do Red Hat são inclusos nos nomes dos arquivos de pacotes e algumas vezes possuem um número de correção de bug interno ou número de correção anexado. Por exemplo, o pacote **bind-9.1.0-10.arch.rpm** é a versão 10 do Red Hat da versão 9.1.0 de isc.org. Confuso.

O Debian possui uma política parecida à do Red Hat. Por vezes, atualizações de segurança liberadas por novos pacotes são adaptadas às versões antigas, presentes na distribuição estável para que não seja incluso um pacote inteiramente novo na distribuição.

Componentes de BIND

O sistema BIND possui três componentes:

- Um daemon chamado **named** que responde a consultas
- Rotinas de biblioteca que solucionam consultas host contactando os servidores do banco de dados DNS distribuído
- Interfaces de linhas de comandos para o DNS: **nslookup**, **dig** e **host**

No linguajar DNS, um daemon como **named** (ou a máquina na qual ele é executado) é chamado de "servidor de nomes", e o código de cliente que o contacta é chamado de "solucionador". Discutimos brevemente a função de cada componente a seguir porém adiamos a real configuração de BIND até a página 282.

named: o servidor de nomes BIND

named responde a consultas sobre nomes de hosts e endereços IP. Se **named** não souber a resposta para uma consulta, ele questiona outros servidores e coloca suas respostas em cache. **named** também executa "transferências de zonas" para copiar dados entre servidores de um domínio. (Uma "zona" é um domínio menos seus subdomínios. Os servidores de nomes lidam com zonas, porém "domínio" é usado onde normalmente se quer dizer "zona".)

Os servidores de nomes podem operar em diversos modos diferentes. As distinções entre eles recaem sobre vários eixos, de modo que a categorização final normalmente não é muito ordenada. Para tornar as coisas ainda mais confusas, um único servidor é capaz de desempenhar diversos papéis em relação a diferentes zonas. A Tabela 16.5 lista alguns dos adjetivos usados para descrever os servidores de nomes. Entradas recuadas são classificadas livremente sob seus cabeçalhos não recuados.

Tabela 16.5 Uma taxonomia de servidores de nomes.

Tipo de servidor	Descrição
autorizado	Um representante oficial de uma zona
mestre	O servidor principal de uma zona; obtém dados de um arquivo em disco
escravo	Copia seus dados do mestre (também chamado de servidor secundário)

7. O release r1 (3.0r1) do Debian distribui a versão 8.3.3 do BIND.

Snow666

Tabela 16.5 Uma taxonomia de servidores de nomes. (*continuação*)

Tipo de servidor	Descrição
stub	Similar a um escravo, porém copia somente dados de servidor de nomes (e não dados de host)
distribuição	Um servidor que é visível[a] somente de dentro de um domínio; (também conhecido como "servidor furtivo")
não autorizado[b]	Responde a uma consulta no cache; não sabe se os dados ainda são válidos
caching	Coloca em cache dados de consultas anteriores; normalmente não possui zonas locais
encaminhador	Executa consultas em nome de diversos clientes; constrói um grande cache
recursivo	Consultas em seu nome até ele retornar uma resposta ou um erro
não recursivo	Refere-se a um outro servidor caso ele não consiga responder a uma consulta

a. Um servidor de distribuição pode ser visível a qualquer um que conheça o seu endereço IP.
b. Estritamente falando, "não autorizado" é um atributo de uma resposta a uma consulta DNS, não um servidor.

Essas categorizações se baseiam na fonte de dados de um servidor de nomes (autorizado, caching, mestre, escravo), no tipo de dados salvos (stub), no caminho de consulta (encaminhadores), no tipo de respostas manipuladas (recursivos, não recursivos) e, finalmente, na visibilidade do servidor (distribuição). As próximas seções fornecem alguns detalhes adicionais sobre as diferenças mais importantes; as demais diferenças são descritas em outros pontos deste capítulo.

Servidores autorizados e somente para caching

Servidores mestre, escravo e somente para caching são distinguidos por duas características: de onde provêm os dados e se o servidor é autorizado para o domínio.

Cada zona possui um servidor de nomes mestre. O servidor mestre mantém a cópia oficial dos dados da zona em disco. O administrador de sistemas altera os dados da zona editando os arquivos de dados do servidor mestre.

Consulte a seção 16.12 para obter maiores informações sobre transferências de zonas.

Um servidor escravo obtém seus dados do servidor mestre através de uma operação de "transferência de zonas". Uma zona pode ter vários servidores de nomes escravos e deve *obrigatoriamente* ter pelo menos um. Um servidor stub é um tipo especial de escravo que carrega somente os registros NS (Name Server, servidor de nomes) do servidor mestre. Consulte a seção 16.11 para obter uma explicação do porquê você poderia desejar esse tipo de comportamento. Uma máquina pode ser ao mesmo tempo um servidor mestre para suas zonas e um servidor escravo para outras zonas. Tal cooperação normalmente gera bons vizinhos DNS.

Um servidor de nomes somente para caching carrega os endereços dos servidores para o domínio-raiz a partir de um arquivo de inicialização e acumula o restante de seus dados colocando dados em cache para as consultas que ele soluciona. Um servidor de nomes somente para caching não possui nenhum dado próprio e não é autorizado para nenhuma zona. Veja *Um departamento de uma universidade* na seção 16.10 para ter um exemplo de uma configuração somente para caching.

Uma resposta autorizada de um servidor de nomes é "garantidamente" acurada; uma resposta de um servidor não autorizado pode estar desatualizada. Entretanto, uma porcentagem muito grande das respostas desse tipo de servidor é perfeitamente correta. Servidores mestre e escravo são autorizados para suas próprias zonas, mas não para informações que eles tenham jogado em cache acerca de outros domínios.

Verdade seja dita, mesmo respostas autorizadas podem ser imprecisas caso um administrador de sistemas altere os dados do servidor mestre e se esqueça de atualizar o número serial dos dados ou de executar **ndc reload** ou **rndc reload** (ou se as alterações ainda não tiverem sido propagadas para os servidores escravos).

O servidor mestre deve ser localizado numa máquina que seja estável, não possua muitos usuários e que talvez esteja conectada a um no-break. Deve haver ao menos dois escravos, um dos quais fora do site. Escravos no site devem residir em redes diferentes e em circuitos de alimentação diferentes. Quando o serviço de nomes pára, todo o acesso normal à rede também pára.

Os dados de zonas de um domínio normalmente incluem as identidades dos servidores de nomes de todos os seus subdomínios. Esse servidor de nomes permite que clientes DNS percorram a árvore de domínios para procurar qualquer host na Internet. Se um domínio-pai não mencionar determinados servidores de nomes de um subdomínio, esses servidores se tornam servidores "internos" e não podem ser acessados externamente.

Embora não sejam autorizados, os servidores somente para caching podem reduzir a latência vista por seus usuários e a quantidade de tráfego DNS em suas redes internas. Considere a possibilidade de colocar um servidor

www.snow666.blogspot.com
Snow666

somente para caching em cada sub-rede. Na maioria dos sites, as máquinas desktop normalmente passam por um servidor de caching para solucionar consultas sobre hosts na Internet.

No BIND 4 e no BIND 8, não era uma boa idéia usar um único servidor de nomes como um servidor autorizado para algumas zonas e como servidor de caching para outras. Cada **named** era executado com um único banco de dados em memória e poderia ocorrer contaminação cruzada caso a memória fosse reduzida e os dados em cache misturados com dados autorizados. O BIND 9 eliminou esse problema, de modo que é possível fazer a mistura.

Servidores recursivos e não recursivos

Os servidores de nomes são recursivos ou não recursivos. Se um servidor não recursivo tiver a resposta a uma consulta colocada em cache de uma transação anterior ou for autorizado para o domínio ao qual pertence a consulta, ele fornece uma resposta apropriada. Caso contrário, em vez de retornar uma resposta verdadeira, ele retornará uma referência aos servidores autorizados de um outro domínio que têm maior probabilidade de conhecer a resposta. Um cliente de um servidor não recursivo tem de ser preparado para aceitar e agir baseado em referências.

Embora os servidores não-recursivos possam parecer preguiçosos, eles normalmente possuem uma boa razão para não pegarem trabalho extra. Os servidores-raiz e de domínio de primeiro nível são todos não-recursivos, porém a 10.000 consultas por segundo podemos desculpá-los por usar a lei do menor esforço.

Um servidor recursivo retorna somente respostas reais ou mensagens de erros. Ele próprio segue referências, liberando o cliente dessa responsabilidade. O procedimento básico para solucionar uma consulta é praticamente o mesmo; a única diferença é que o servidor de nomes cuida da manipulação de referências em vez de passá-las de volta ao cliente.

As bibliotecas solucionadoras *não* compreendem referências; o servidor de nomes local que os clientes indicam deve ser recursivo.

Um efeito colateral de fazer com que um servidor de nomes siga referências é que o seu cache adquire informações sobre domínios intermediários. Numa rede local, esse caching normalmente é o comportamento desejado, já que ele permite que pesquisas subseqüentes em qualquer host da rede se beneficiem do trabalho anterior do servidor de nomes. Por outro lado, o servidor para um domínio de primeiro nível, como com ou edu, não deve salvar informações solicitadas por um host vários domínios abaixo dele.

As primeiras versões de BIND exigiam mudanças no código-fonte e uma recompilação para modificar a recursividade de um servidor. Essa opção então foi movida para um flag (**-r**) de linha de comando e agora é um parâmetro no arquivo de configuração. Um servidor pode até mesmo ser configurado como recursivo para seus próprios clientes e não recursivo para os de fora.

Os servidores de nomes geram referências de maneira hierárquica. Por exemplo, se um servidor não for capaz de fornecer um endereço para o host lair.cs.colorado.edu, ele se refere aos servidores de cs.colorado.edu, colorado.edu, edu ou ao domínio-raiz. Uma referência tem de incluir endereços para os servidores dos domínios referidos, de modo que a escolha não é arbitrária, o servidor tem de se referir a um domínio para o qual ele já conhece os servidores.

Normalmente é retornado o domínio mais longo conhecido. Se o endereço de lair não fosse conhecido porém os servidores de nomes para cs.colorado.edu fossem conhecidos, então seriam retornados os endereços para estes servidores. Se cs.colorado.edu não fosse conhecido mas colorado.edu sim, então os endereços dos servidores de nomes para colorado.edu seriam retornados e assim por diante.

Os servidores de nomes carregam previamente seus caches por meio de um arquivo de "dicas" que lista os servidores para o domínio-raiz. Sempre pode ser feita alguma referência, mesmo que simplesmente: "Vá perguntar a um servidor-raiz".

A biblioteca solucionadora

Os clientes procuram associações de nomes de hosts chamando a família de rotinas de bibliotecas **gethostbyname**. A implementação original de **gethostbyname** pesquisava nomes no arquivo **/etc/hosts**. Para que associações de hosts sejam fornecidas pelo DNS, estas rotinas têm de usar a biblioteca solucionadora, que sabe como localizar e comunicar-se com os servidores de nomes.

O Linux integrou o solucionador nas bibliotecas-padrão. As implementações de **gethostbyname** feitas pelos sistemas Linux podem extrair informações de várias fontes diferentes: arquivos planos (como **/etc/hosts**), DNS e talvez um sistema de banco de dados administrativo local como o NIS. Um arquivo de chaves possibilita um controle administrativo detalhado sobre quais fontes são pesquisadas e em que ordem. Consulte a seção 18.3, ou a seção 16.16, mais à frente para obter detalhes.

Snow666

O BIND 9 contém uma biblioteca solucionadora leve, que permite aos sites, os quais não precisam falar IPv6, usar uma biblioteca solucionadora enxuta juntamente com um daemon solucionador, **lwresd**. Se a instrução `lwres` for incluída no arquivo de configuração de **named**, o servidor de nomes também atuará como um solucionador leve.

Interfaces de shell para o DNS

A distribuição de software BIND inclui os comandos **dig**, **host** e **nslookup**, que fornecem uma interface de linha de comando para execução de consultas DNS. Elas são úteis como ferramentas de depuração e como ferramentas para extração de informações do DNS. Embora os comandos sejam similares em função, eles são um tanto diferentes em termos de design. Consulte a seção 16.14 para obter maiores informações.

16.7 Funcionamento do DNS

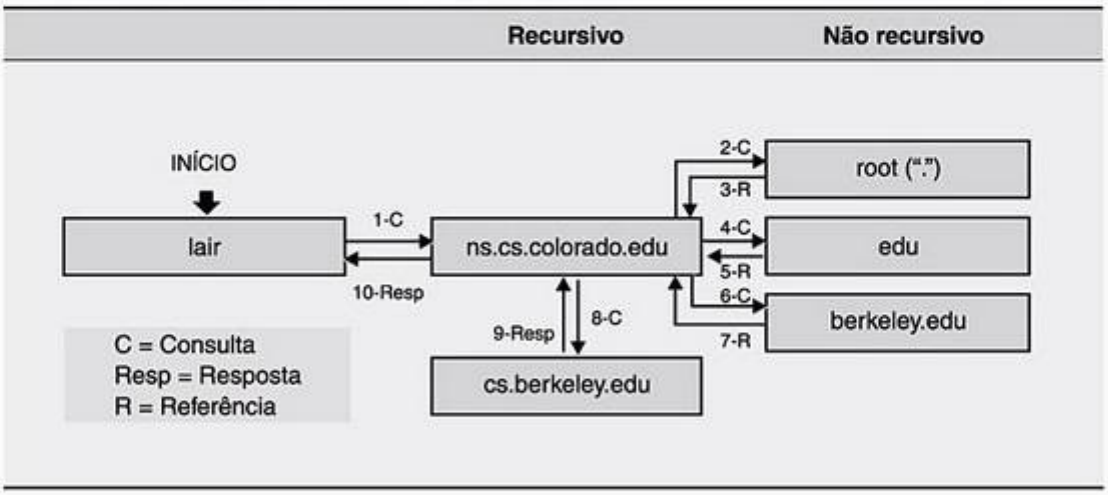
Cada host que usa DNS é um cliente do sistema ou simultaneamente um cliente e um servidor. Caso planeje não usar nenhum servidor DNS, não é essencial que você leia as próximas seções (vá direto para *Configuração do solucionador* na seção16.8). Entretanto, as próximas seções o ajudarão a ter um entendimento mais sólido da arquitetura do DNS.

Delegação

Todos os servidores de nomes têm conhecimento dos servidores-raiz. Por sua vez, os servidores-raiz têm conhecimento dos domínios com, org, edu, fi, de e outros domínios de primeiro nível. Mais abaixo na cadeia, edu tem conhecimento de colorado.edu, com sabe da existência de admin.com e assim por diante. Cada zona é capaz de delegar autoridade sobre seus subdomínios para outros servidores.

Vejamos um exemplo real. Suponha que queiramos pesquisar o endereço da máquina vangogh.cs.berkeley.edu na máquina lair.cs.colorado.edu. O host lair solicita seu servidor de nomes local, ns.cs.colorado.edu, para encontrar a solução. O Diagrama A ilustra os eventos subseqüentes. Usamos nomes relativos para despoluir um pouco as coisas e fazer com que as legendas ficassem mais legíveis. Os números nas setas entre os servidores mostram a ordem dos eventos e uma letra indica o tipo de transação (consulta, referência ou resposta). Fizemos a suposição de que nenhuma das informações necessárias fosse colocada em cache antes da consulta, exceto pelos nomes e endereços IP dos servidores do domínio-raiz.

Diagrama A Processo de consultas DNS para vangogh.cs.berkeley.edu.



O servidor de nomes local não conhece o endereço; além disso, ele não sabe nada a respeito de cs.berkeley.edu ou berkeley.edu ou mesmo de edu. Porém, ele conhece efetivamente alguns servidores do domínio-raiz e, já que é um servidor recursivo, ele consulta um servidor-raiz a respeito de vangogh.cs.berkeley.edu.

Os servidores-raiz costumavam ser usados tanto para conter dados para zonas-raiz quanto para os gTLDs, porém nos dias atuais os gTLDs com, net e org possuem seus próprios servidores. O domínio edu deve realmente possuir seus próprios servidores também, embora, na realidade, ele seja servido no momento pelos servidores-raiz. Para as finalidades deste exemplo, suporemos que recebemos uma referência para alguns servidores independentes do domínio edu em resposta à nossa consulta de servidor-raiz sobre vangogh.cs.berkeley.edu.

Snow666

O servidor de nomes local envia então sua consulta para um servidor edu (perguntando, como sempre, sobre vangogh.cs.berkeley.edu) e obtém como resposta uma referência aos servidores para berkeley.edu. Ele repete então a consulta no domínio berkeley.edu. Se o servidor Berkeley não for recursivo e não tiver a resposta colocada em cache, ele retorna uma referência a cs.berkeley.edu. O servidor cs.berkeley.edu é autorizado para as informações solicitadas e retorna o endereço de vangogh.

Quando a poeira abaixa, ns.cs.colorado.edu coloca em cache o endereço de vangogh. Ele também tem listas de servidores colocadas em cache para edu, berkeley.edu e cs.berkeley.edu.

Caching e eficiência

O caching aumenta a eficiência das pesquisas: uma resposta colocada em cache fica quase que totalmente livre e normalmente é correta, pois os mapeamentos mudam com pouca freqüência. A maior parte das consultas é para hosts locais e podem ser solucionadas rapidamente. Os usuários também ajudam inadvertidamente a eficiência, pois repetem muitas consultas; após a primeira ocorrência de uma consulta, o restante fica "livre".

Por um longo tempo, o caching era aplicado somente a respostas positivas. Se o endereço ou nome de um host não puder ser localizado, este fato não era salvo. Um esquema para caching DNS negativo foi descrito na RFC1034, porém ele era incompleto e não foi implementado na maioria das versões de BIND. Em 1998, a RFC2308 definiu um esquema atualizado para caching negativo. Esse esquema foi implementado no BIND 8.2 como um recurso opcional; ele é obrigatório no BIND 9.

Uma medida no servidor-raiz RIPE na Europa indicava que 60% das consultas DNS eram para dados inexistentes (muitas consultas eram para 127.in-addr.arpa ou para serviços Microsoft como nomes de hosts). Colocar em cache essas informações mais abaixo na árvore DNS deveria reduzir dramaticamente a carga nos servidores-raiz.

Caching negativo salva respostas dos seguintes tipos:

- Nenhum host ou domínio atende ao nome consultado.
- O tipo de dados solicitado não existe para este host.
- O servidor a ser consultado não está respondendo.
- O servidor não pode ser alcançado devido a problemas na rede.

Os dois primeiros tipos de dados negativos são colocados em cache por 1–3 horas e os demais tipos por 5 minutos. Respostas não autorizadas *podem vir a* ser colocadas em cache; respostas negativas autorizadas devem *obrigatoriamente* ser colocadas em cache. O BIND segue essas diretrizes com base nas RFCs, porém os computadores com Windows parecem implementar as TTLs de forma seletiva, pelo menos no que tange ao caching negativo. Eles usam o mínimo do registro SOA na primeira vez que uma consulta retorna NXDOMAIN (no such domain), depois reinicia o TTL para 15 minutos e deixa que ele se expire normalmente a partir daí.

named normalmente recebe vários registros DNS em resposta a uma consulta. Por exemplo, uma consulta para os servidores de nomes do domínio-raiz receberia uma resposta que listava todos os 13 servidores-raiz. Qual deles seu servidor deveria consultar?

Quando **named** tem de optar entre vários servidores remotos, todos os quais autorizados para um dado domínio, ele determina primeiramente o RTT (Round Trip Time, tempo de percurso ida e volta) para cada servidor. A seguir ele ordena os servidores em "baldes" de acordo com seus RTTs e seleciona um servidor do balde mais rápido. Servidores dentro de um balde são tratados como equivalentes e são usados numa forma round robin.

Você pode obter uma forma primitiva, porém eficiente, de balanceamento de carga atribuindo um único nome de host a vários endereços IP (que, na realidade, são máquinas diferentes):

```
www         IN A  192.168.0.1
            IN A  192.168.0.2
            IN A  192.168.0.3
```

Servidores Web muito solicitados como o Yahoo ou o AltaVista não são, na verdade, uma única máquina. Eles são apenas um único nome no DNS. Um servidor de nomes que possui vários registros para o mesmo nome e tipo de registro retorna todos eles ao cliente, contudo numa ordem round robin. Por exemplo, a ordem round robin para os registros A acima seria 1, 2, 3 para a primeira consulta, 2, 3, 1 para a consulta seguinte, 3, 1, 2 para a terceira e assim por diante.

O protocolo DNS estendido

A definição do protocolo DNS original data do final dos anos 80 e usa tanto UDP quanto TCP na porta 53. O UDP é usado tipicamente para consultas e respostas e o TCP para transferências de zonas entre servidores mestres e servi-

www.snow666.blogspot.com
Snow666

dores escravos. Infelizmente, o tamanho máximo do pacote que garantidamente funcionará em todas as implementações do UDP é de 512 bytes, que é muito pequeno para alguns dos novos recursos de segurança DNS que devem incluir assinaturas digitais em cada pacote.

A restrição de 512 bytes também afeta o número e os nomes dos servidores-raiz. Para fazer com que todos os dados de servidor-raiz caibam num pacote UDP de 512 bytes, o número de servidores-raiz é limitado a 13, e cada servidor é nomeado apenas com uma única letra do alfabeto.

Muitos solucionadores enviam primeiramente uma consulta UDP, depois, se receberem uma resposta truncada, eles enviam novamente a consulta via TCP. Esse procedimento contorna o limite de 512 bytes, porém, é ineficiente. Você poderia imaginar que o DNS deveria simplesmente descartar o UDP e usar o TCP a todo o momento, porém as conexões TCP são muito mais caras. Uma troca de informações de servidor de nomes com UDP pode ser tão curta quanto dois pacotes, um de consulta e um de resposta. Uma troca de informações TCP envolve pelo menos sete pacotes: um handshake de três vias para iniciar a conversação, uma consulta, uma resposta e um handshake final para fechar a conexão.

Na metade dos anos 90, o protocolo DNS foi reformulado para incluir transferências incrementais de zonas (como um **diff** entre arquivos de zonas antigos e novos, inspirado no programa **patch** de Larry Wall), notificações assíncronas (para informar aos escravos quando os arquivos de dados do mestre tinham sido atualizados) e atualizações dinâmicas (para hosts DHCP). Essas modificações adicionaram recursos, mas não resolveram realmente a questão fundamental de transporte.

No final dos anos 90, o EDNS0 (Extended DNS, versão 0) solucionou alguns dos problemas do protocolo DNS na Internet atual. Ele permite aos transmissores anunciarem tamanho de buffer remontado, opções suportadas e versões de protocolo compreendidas. Se o servidor de nomes responder com uma mensagem de erro, o emissor retorna ao protocolo DNS original. O BIND 9 implementa EDNS0 tanto no servidor quanto no solucionador.

16.8 Questões Relacionadas a Clientes no BIND

Antes de mergulharmos de cabeça na configuração do BIND, enumeremos as tarefas associadas ao emprego do BIND na Internet. A Tabela 16.6 sintetiza o que deve ser feito, para quem e com que freqüência. Uma entrada na coluna "Com que freqüência" que inclua a palavra "distribuir" significa que você o faz uma vez só por sub-rede ou arquitetura e depois copia o resultado para os hosts apropriados com uma ferramenta do tipo **rdist** ou **rsync**. É aplicado um sombreamento em linhas alternadas para facilitar a leitura.

Veja o Capítulo 18 para obter maiores informações sobre distribuição de arquivos numa rede.

Tabela 16.6 Tarefas de instalação e de manutenção no BIND.

Tarefa	Para quem	Com que freqüência
Obter o nome do domínio	Site	Somente uma vez
Escolher servidores de nomes	Site	Somente uma vez
Obter distribuição BIND	Site	Somente uma vez, mas mantê-la atualizada
Configurar solucionador	Cliente	Somente uma vez e distribuir
Configurar solucionador eficiente	Cliente	A cada sub-rede e distribuir
Configurar comutação de serviços	Cliente	A cada arquitetura e distribuir
Iniciar **named** no momento do boot	Servidor	A cada servidor de nomes
Configurar o arquivo de configuração **named**	Servidor	A cada tipo de servidor
Configurar o arquivo de dicas	Servidor	Somente uma vez[a] e distribuir aos servidores
Configurar arquivos de zonas	Mestre	Somente uma vez
Atualizar arquivos de zonas	Mestre	Conforme necessário
Revisar arquivos de log	Host de log	Pelo menos uma vez por semana
Instruir os usuários	Todos os hosts	Contínua e repetidamente

a. Mas tem de ser refeita caso os servidores-raiz mudem.

Já que cada host da rede tem de ser um cliente BIND, comecemos nossa discussão detalhada com tarefas do lado cliente.

www.snow666.blogspot.com
Snow666

Configuração do solucionador[8]

Cada host da rede possui um arquivo chamado **/etc/resolv.conf** que lista os servidores DNS que o host deve consultar. Se seu host obtiver o seu endereço IP e parâmetros de rede de um servidor DHCP, o arquivo **/etc/resolv.conf** deve ser configurado para você de forma automática. Caso contrário, você terá que editá-lo manualmente. O formato é:

```
search nome-de-domínio …
nameserver endereço-IP
```

Podem ser listados até três servidores de nomes. Eis um exemplo completo:

```
search cs.colorado.edu colorado.edu ee.colorado.edu
nameserver 128.138.243.151  ; ns
nameserver 128.138.204.4    ; piper
nameserver 128.138.240.1    ; anchor
```

Jamais foram definidos comentários para o arquivo **resolv.conf**. Eles são relativamente suportados no sentido em que qualquer coisa que não seja reconhecida será ignorada. É seguro colocar comentários no final de linhas de servidor de nomes, pois o analisador sintático somente procura um endereço IP e ignora o restante da linha. Pelo fato de a linha de busca poder conter vários argumentos, comentários ali poderiam provocar problemas.

A linha `search` lista os domínios a serem consultados caso um nome de host não seja totalmente qualificado. Se, por exemplo, um usuário emitir o comando **ssh foo**, o solucionador completa o nome com o primeiro domínio da lista de pesquisa (no arquivo **resolv.conf** acima, cs.colorado.edu) e procura foo.cs.colorado.edu. Se tal nome não puder ser encontrado, o solucionador também tenta foo.colorado.edu e foo.ee.colorado.edu.

Usuários em nosso subdomínio cs podem usar nomes de hosts simples para qualquer host local, mas os usuários no domínio-pai devem usar *nome_do_host*.cs para atingir um host no subdomínio. Se você criar subdomínios, também terá de reconfigurar (instruir) seus usuários.

Uma diretiva de busca nos arquivos **resolv.conf** das máquinas no domínio-pai poderia permitir o emprego de nomes de hosts simples em ambas as direções:

```
search colorado.edu. cs.colorado.edu. ee.colorado.edu.
```

Obviamente, essa configuração supõe que os nomes de host sejam exclusivos ao longo dos três domínios. Uma diretiva de busca pode especificar até oito domínios.

Os servidores listados em **resolv.conf** têm de ser recursivos (já que o solucionador não entende referências), e cada um deles deve ter um cache. Se você estiver usando BIND 4 ou BIND 8, os servidores não devem ser autorizados para qualquer zona. Seus caches podem crescer em demasia, e como as versões 4 e 8 não gerenciam o cache apropriadamente, pode ser que toda a memória da máquina seja consumida. Se tiver de mesclar dados em cache e autorizados, veja a opção de configuração `listen-on` na seção 16.9 para obter uma maneira segura de fazê-lo — executando, na mesma máquina, dois servidores distintos que escutam portas diferentes.

Os servidores nas linhas `nameserver` são contactados em ordem; enquanto o primeiro deles continuar a responder consultas, os demais serão ignorados. Se ocorrer um problema, a consulta é expirada e será tentado o próximo servidor de nomes. Cada servidor é tentado alternadamente, até quatro vezes. O intervalo de timeout aumenta a cada falha. Quando o DNS está com problemas, parece que toda a rede está problemática.

A maioria dos solucionadores permite que sejam listados no máximo três servidores de nomes. Se for fornecido um número maior, eles serão silenciosamente ignorados. Se o próprio host for um servidor de nomes, ele deve ser listado primeiramente em seu próprio arquivo **resolv.conf**.

As primeiras versões do BIND usavam a diretiva `domain` no arquivo **resolv.conf** em vez da diretiva `search`. Ela especificava um único domínio para adicionar nomes que não eram totalmente qualificados. Recomendamos substituir as diretivas `domain` por diretivas `search`. As duas diretivas são mutuamente exclusivas, de modo que somente uma deve estar presente. Se você tiver um solucionador mais antigo e incluir ambas as diretivas no arquivo **resolv.conf**, será usada a última delas.

Sob uma perspectiva interplataforma, o comportamento-padrão dos solucionadores atuais é uma verdadeira confusão. Alguns são mais agressivos com valores-padrão do que outros. Alguns supõem que a máquina local seja um servidor DNS caso não seja listado nenhum servidor de nomes. Alguns desconstroem um nome de host local (totalmente qualificado) para gerar uma lista de busca. Outros são capazes de operar sem nenhum arquivo

8. O Linux usa um arquivo "switch" (**/etc/nsswitch.conf** ou, para aplicativos muito antigos, **/etc/host.conf**) que especifica que fontes de dados devem ser usadas para implementar pesquisas de nomes. Cada uma de nossas distribuições especifica DNS, embora nem sempre em primeiro lugar.

www.snow666.blogspot.com
Snow666

/etc/resolv.conf em si. Não conte com nenhum desses recursos distorcidos. Simplesmente configure **resolv.conf** normalmente para cada um de seus hosts, independentemente da versão de UNIX ou Linux que estiver usando.

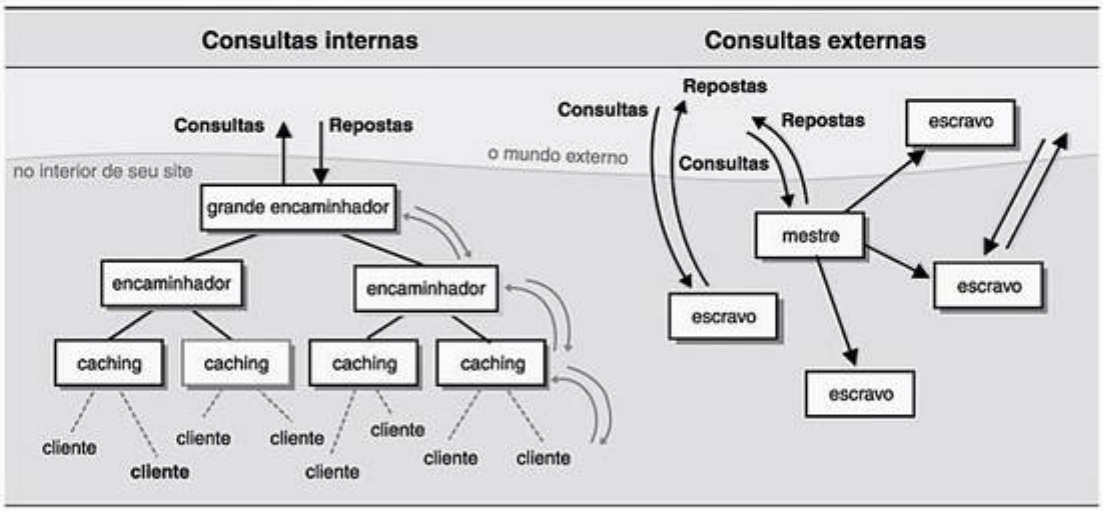
As consultas DNS que chegarem do mundo externo virão para seus servidores de nomes autorizados. É uma boa idéia dispor de servidores distintos para solucionar consultas de dentro de seu domínio. Os seus servidores internos devem ser recursivos e somente para cache. Um site grande deve ter vários servidores de nomes e deve personalizar o arquivo **resolv.conf** para distribuir a carga entre os servidores, minimizar o tráfego de rede e reduzir a vulnerabilidade das máquinas para um único ponto de falha. Se o serviço de nomes for interrompido, a sua instalação inteira será paralisada.

Encaminhadores também se constituem numa boa maneira para um site local otimizar serviço de nomes. Os servidores de nomes locais apontam para um encaminhador que produz todas as consultas externas para o seu site e constrói um cache muito rico. Esta configuração minimiza a largura de banda externa usada para serviço de nomes e permite que todas as máquinas locais compartilhem um único e grande cache. Os encaminhadores são vistos no item sobre configuração que começa na seção 16.9.

O Diagrama B ilustra a estrutura recomendada nos parágrafos anteriores. Ele mostra uma hierarquia de encaminhamento em dois níveis, que é demasiado para a maioria dos sites. Ajuste o balanço entre os servidores que manipulam consultas que vão para fora e os servidores que atendem a consultas que chegam de fora de modo que nenhum dos grupos seja muito sobrecarregado.

Note também o emprego do servidor escravo fora do site, que é altamente recomendado. Freqüentemente, um ISP, ou universidade local, estará disposto a rodar um servidor fora de sua instalação para o seu domínio. Empresas com vários escritórios podem designar uma localidade como mestre; esta localidade se torna "fora do site" em relação ao restante da empresa.

Diagrama B Arquitetura de servidor DNS



Testes de solucionadores

Em sistemas Linux, tudo o que temos a fazer para começar a usar o DNS é acrescentar uma linha de servidor de nomes ao arquivo **/etc/resolv.conf**. O arquivo de chaves padrão (**/etc/nsswitch.conf**) já inclui `dns`; veja a página 373 para obter maiores informações.

Após configurar **/etc/resolv.conf** (e supondo que sua conexão de rede local esteja ativa e funcionando corretamente), você deve ser capaz de se referir a outras máquinas pelo nome em vez de seu endereço IP. Se você tentar atingir uma outra máquina local e o comando simplesmente travar, tente se referir à máquina pelo seu endereço. Se isso funcionar, então sua configuração DNS é o problema. Verifique se os endereços IP de servidor de nomes no arquivo **/etc/resolv.conf** estão corretos e se os servidores que você indicou permitem consultas em sua rede (seção 16.9).

Impacto sobre o restante do sistema

A mudança de tabelas de hosts estáticas para DNS cria algumas dependências potenciais na inicialização e configuração contra as quais você deve se proteger. Quando um host faz sua inicialização, pode ser que referências a nomes e hosts presentes nos arquivos de inicialização sejam insolúveis caso eles sejam encontrados antes da rede estar ativa.

www.snow666.blogspot.com
Snow666

Os comandos nos arquivos de inicialização tentarão, em vão, contactar o DNS. Graças à robustez do solucionador, eles tentarão várias vezes em vários servidores distintos, aumentando o respectivo período de timeout a cada tentativa frustrada. Alguns minutos depois, o comando que precisava do nome de host, finalmente falhará.

Para solucionar o problema, use somente endereços IP literais nos estágios iniciais do processo de boot.

Ou então, já que o Linux suporta o emprego simultâneo de DNS e de **/etc/hosts** por meio do arquivo de chaves, você poderá instalar um arquivo **hosts** contendo os endereços de servidor necessários no momento do boot. Certifique-se de que o arquivo **hosts** seja checado antes do DNS de forma a não ter que esperar o DNS atingir seu período de timeout.

16.9 Configuração de Servidores BIND

Nesta seção, partimos da suposição de que suas tarefas políticas foram completadas. Isto é, supomos que você tenha um nome de domínio (possivelmente um subdomínio), ajustou-se com o administrador DNS do domínio-pai e recebeu autoridade sobre seu espaço de endereços na árvore invertida in-addr.arpa. Você já escolheu o nome de seu servidor de nomes mestre e uma série de secundários e já instalou o BIND.

Exigências de Hardware

O BIND é um consumidor voraz de memória. Seu banco de dados é mantido em memória, de modo que à medida que o cache cresce, o mesmo acontece com o processo **named**. Alguns dos novos recursos de BIND 9 também consomem muitos recursos de CPU, mais notadamente DNSSEC e IPv6. Para ajudar a reduzir toda essa carga, o BIND 9 é multithread e é capaz de fazer pleno uso dos sistemas multiprocessados. O BIND 9 também contém opções de configuração que controlam o emprego de recursos por parte de **named**.

A melhor maneira de determinar se um servidor de nomes possui memória suficiente é executá-lo por um instante e observar o tamanho do processo **named**. Leva de uma a duas semanas até se chegar a um tamanho estável no qual registros de cache antigos devem expirar praticamente na mesma taxa que os novos estão sendo inseridos. Uma vez estabilizado, o sistema não deve fazer swapping e suas taxas de paginação devem ser razoáveis.

Se o seu servidor de nomes é executado numa máquina dedicada, uma ótima regra é a máquina ter o dobro de memória consumido por **named** depois de ele ter sido executado por duas semanas. Os comandos **top** e **vmstat** mostram o emprego de memória; veja *Analisando o emprego de memória*, na página 586 para obter maiores detalhes.

Inicialização de named

named é inicializado no momento do boot e é executado continuamente. Cada uma das distribuições Linux adotadas como exemplo neste livro possui um script de inicialização para **named** que é executado por meio de **init**: **/etc/init.d/named** no Red Hat e SuSE e por meio de **/etc/init.d/bind** no Debian.

Para controlarmos uma cópia em execução de **named**, usamos a interface de linha de comando chamada **ndc** no BIND 8 e **rndc** no BIND 9. **ndc** e **rndc** são descritas de forma mais detalhada no tópico referente à depuração no BIND que se inicia na seção 16.14. O emprego é simplesmente

```
# ndc command
```

Alguns comandos úteis são **start**, **stop**, **restart** e **status**, que possuem significados evidentes. Não tivemos sorte com **ndc start** em sistemas Red Hat; em vez dele tente o seguinte comando:

```
# /sbin/service named start
```

Os scripts **init** que inicializam **named** em cada sistema fornecem alguns pontos de entrada adicionais (como **reload**) que são destinados a serem empregados por administradores de sistemas. Entretanto, a implementação em algumas distribuições é questionável e é mais fácil usar **ndc** de qualquer maneira. Sugerimos que você deixe os scripts de inicialização para **init** — nenhuma necessidade de criar dependências de sistema desnecessárias em seus scripts administrativos ou em seu cérebro.

*Consulte a seção 28.3 para obter maiores informações sobre **inetd**.*

named usa syslog e, conseqüentemente, syslogd deve ser inicializado antes de **named**. Não use **inetd/xinetd** para controlar **named**; ele irá reiniciar **named** cada vez que for necessário, aumentando os tempos de resposta e impedindo que qualquer cache útil seja desenvolvido.

Snow666

Arquivos de configuração

A configuração completa para **named** é formada pelo arquivo de configuração, o arquivo de dicas e, para servidores mestres, os arquivos de dados de zonas contendo os mapeamentos de endereços para cada host. O arquivo de configuração de **named** possui o seu formato próprio; todos os demais arquivos são conjuntos de registros de dados DNS formatados de acordo com a especificação DNS. Discutimos o arquivo de configuração nas próximas duas seções. O formato de registros de dados DNS é descrito na seção 16.11.

O arquivo de configuração de **named** especifica o papel (mestre, escravo ou stub) deste host em relação a cada zona e a maneira pela qual ele deve obter sua cópia dos registros de recursos que formam a parte local do banco de dados. Ele também é o local onde as opções são especificadas — tanto opções globais relacionadas à operação de **named** em si quanto opções específicas a servidores ou zonas que afetam determinados locais.

O formato do arquivo de configuração de **named** mudou completamente entre o BIND versão 4 e o BIND 8, com o novo formato lembrando aquele de **gated.conf** (veja a página 237). O nome do arquivo de configuração também mudou: no BIND 4, ele era **/etc/named.boot** e no BIND 8 e 9 ele é **/etc/named.conf**. Os arquivos de cache e arquivos de dados permanecem os mesmos.

Descrevemos o arquivo de configuração do BIND 8/9 e ignoramos completamente o BIND 4. Como acontece com qualquer software, versões mais antigas do BIND podem conter vulnerabilidades em termos de segurança que foram corrigidas nas versões atuais.

O arquivo **named.conf** é formado por uma série de instruções, cada uma das quais terminada por um ponto-e-vírgula. Identificadores são separados por espaços em branco, que podem incluir linhas novas. As chaves são algumas vezes usadas para agrupamento, mas somente em contextos específicos. O formato é bastante frágil — a simples ausência de um ponto-e-vírgula pode gerar problemas.

Felizmente, o BIND 9 inclui uma série de ferramentas úteis para verificar a sintaxe do arquivo de configuração (**named-checkconf**) e os arquivos de dados de zonas (**named-checkzone**). Eles saem em busca de erros e omissões. Por exemplo, **named-checkzone** irá lhe informar se você se esqueceu de incluir uma diretiva `$TTL`. Infelizmente ele não captura tudo. Por exemplo, registros de adesão faltantes (veja a seção 16.11) não são informados e provocam cargas pesadas nos servidores-raiz e gTLD.

Comentários podem aparecer em qualquer ponto em que o espaço em branco for indicado. São válidos e compreendidos comentários em C, C++ e tipo shell:

```
/* Isto é um comentário e pode abranger linhas. */
// Tudo até o final da linha é um comentário.
# Tudo até o final da linha é um comentário.
```

mas você deve escolher um estilo e usá-lo de maneira consistente.

Cada instrução começa com uma palavra-chave que identifica o tipo de instrução. Pode haver mais do que uma instância de cada tipo de instrução, exceto para opções e registro. Instruções e partes de instruções também podem ser omitidas, invocando comportamento-padrão para os itens faltantes. A Tabela 16.7 lista as instruções disponíveis. Linhas alternadas foram sombreadas para aumentar a legibilidade.

Tabela 16.7 Tipos de instruções usados em named.conf (BIND 8 e 9).

Instrução	Função
include	Interpola um arquivo (por exemplo, chaves confiáveis legíveis somente por **named**)
options	Configura opções e padrões de configuração de servidor de nomes globais
server	Especifica opções por servidor
lwres	Configura o servidor de nomes como um solucionador leve também (BIND 9 somente)
key	Define informações de autenticação
acl	Define listas de controle de acesso
zone	Define uma zona de registros de recursos
trusted-keys	Usa chaves preconfiguradas
controls	Define canais usados para controlar o servidor de nomes com **ndc**
logging	Especifica categorias de registro e seus destinos
view	Dá uma visão do espaço de nomes (somente para BIND 9)

Snow666

Antes de descrevermos essas instruções e a maneira pelas quais elas são usadas para configurar **named**, precisamos descrever uma estrutura de dados que é usada em muitas das instruções: a lista de concordância de endereços. Uma lista desse tipo é uma generalização de um endereço IP que pode incluir os seguintes itens.

- Um endereço IP (por exemplo, 199.165.145.4)
- Uma rede IP especificada com uma máscara de rede CIDR (por exemplo, 199.165/16)
- O nome de uma lista de controle de acesso definida previamente (veja a seção16.9)
- Uma chave de autenticação criptográfica
- O caractere! para negar coisas

As listas de concordância de endereços são usadas como parâmetros para várias instruções e opções. Eis alguns exemplos:

```
{ ! 1.2.3.13; 1.2.3/24; };
{ 128.138/16; 198.11.16/24; 204.228.69/24; 127.0.0.1; };
```

A primeira dessas listas exclui o host 1.2.3.13 mas permite o restante da rede 1.2.3/24; uma segunda define as redes atribuídas à Universidade de Colorado. As chaves e o ponto-e-vírgula final realmente não fazem parte da lista de concordância de endereços, mas fazem parte da instrução de fechamento que os utiliza.

Quando uma rede ou endereço IP é comparado a uma lista de concordância, a lista é pesquisada em ordem até que seja encontrada uma concordância. Este algoritmo de "primeira concordância" torna importante o ordenamento das entradas. Por exemplo, a primeira lista de concordância de endereços acima não teria o efeito desejado caso as duas entradas fossem invertidas, pois 1.2.3.13 seria bem-sucedido na concordância com 1.2.3/24, mas a entrada invertida jamais seria encontrada.

Agora, vamos às instruções! Algumas são curtas e doces; outras quase que precisam de um capítulo inteiro só para elas.

A instrução include

Para subdividir ou organizar melhor uma configuração grande, você pode colocar diferentes trechos da configuração em arquivos separados. Arquivos subsidiários são trazidos para o arquivo **named.conf** pela instrução `include`:

```
include "caminho";
```

Se *caminho* for relativo, então ele será interpretado como relativo em relação ao diretório especificado na opção `directory`. Um emprego comum da instrução `include` é o de trazer chaves criptográficas que não deveriam ser legíveis por todos. Em vez de fechar o acesso de leitura ao arquivo **named.conf** inteiro, alguns sites mantêm chaves em arquivos com permissões restritas que somente **named** é capaz de ler. Esses arquivos são então incluídos em **named.conf**.

A instrução options

A instrução `options` especifica opções globais, algumas das quais podem ser posteriormente anuladas para determinadas zonas ou servidores. O formato genérico é:

```
options {
        opção;
        opção;
        ...
};
```

Se não estiver presente nenhuma instrução `options` em **named.conf**, então serão usados valores-padrão. O BIND 8 possui cerca de 30 opções e o BIND 9 mais de 50. Para uma lista completa, refira-se à documentação do BIND ou ao livro *DNS and BIND* da O'Reilly de Paul Albitz e Cricket Liu (a terceira edição cobre o BIND 8 e a quarta, as versões 8 e 9). Reduzimos a cobertura dessas opções e discutimos somente aquelas cujo uso recomendamos. (Também solicitamos sugestões por parte dos desenvolvedores do BIND sobre quais opções deveríamos focar e aceitamos os seus conselhos.) Os valores-padrão são enumerados entre colchetes ao lado de cada opção.

```
version "string";                    [número real de versão do servidor]
```

Há duas linhas de pensamento sobre a questão de ocultar o número da versão. Alguns sites acreditam que seus servidores serão mais vulneráveis a ataques caso os hackers consigam saber que versão está sendo executada neles.

www.snow666.blogspot.com
Snow666

Outros acreditam que ocultar as informações sobre versão é contraproducente, pois os hackers tentarão a sorte de qualquer maneira, e a maioria dos bugs recentemente descoberta está presente em todas as versões do software.

Recomendamos que você não reinicie a string de versão apesar do nosso site de exemplo fazer isso (perdemos o argumento da segurança-pela-obscuridade com o grupo sysadmin). É muito prático ser capaz de consultar seus servidores de nomes e descobrir que versão eles estão rodando (por exemplo, no caso de você querer saber se o seu fornecedor está lhe entregando uma versão atual ou se você precisar verificar se de fato atualizou todos os seus servidores para a última revisão).

```
directory "caminho";                    [diretório no qual o servidor foi inicializado]
```

A instrução `directory` faz com que **named** mude para o diretório (**cd**) especificado. Sempre que nomes de caminhos relativos aparecerem nos arquivos de configuração de **named**, eles serão interpretados de forma relativa a este diretório. O caminho (*caminho*) deve ser um caminho absoluto. Quaisquer arquivos de saída (depuração, informações estatísticas etc.) também são gravados nesse diretório.

Recomendamos colocar todos os arquivos de configuração relacionados com BIND (além de **named.conf** e **resolv.conf**) num subdiretório abaixo de **/var** (ou seja lá onde você mantiver seus arquivos de configuração para outros programas). Usamos **/var/named**.

```
notify yes | no;                        [yes]
also-notify endereços-IP-servidores;    [vazio]
```

Se `notify` for configurado para `yes` e este **named** for o servidor mestre para uma ou mais zonas, **named** notificará automaticamente os servidores escravos destas zonas sempre que o banco de dados de zonas correspondente mudar. Os servidores escravos podem então se encontrar com hora marcada com o mestre para atualizar suas cópias dos dados de zona. A opção `notify` pode ser usada tanto como uma opção global quanto como uma opção específica a uma zona. Ela faz com que os arquivos de zonas convirjam mais rapidamente após você proceder a alterações.

Consulte a seção 16.11 para obter maiores informações sobre zonas stub.

named normalmente descobre quais máquinas são servidores escravos de uma zona analisando os registros NS desta zona. Se `also-notify` for especificada, um conjunto de servidores adicionais que não são anunciados com registros NS também podem ser notificados. Esse ajuste fino algumas vezes é necessário quando seu site tiver servidores internos. Não use a opção `also-notify` em servidores stub; eles estão interessados somente nos registros NS da zona e podem aguardar o ciclo de atualização regular. O alvo da opção `also-notify` tem de ser uma lista de endereços IP — uma `acl` não funcionará.

Servidores BIND 4 não compreendem mensagens de notificação. Eles registram um erro em log e aguardam que o intervalo de renovação prescrito nos dados de zona (seção 16.11) se expire antes de se atualizarem. A zona reversa de host local é um bom lugar para desativar a notificação.

```
recursion yes | no;                                  [yes]
allow-recursion { lista-de-concordância-de-endereços };  [todos-os-hosts]
```

A opção `recursion` especifica se **named** consulta outros servidores de nomes em nome dos clientes, conforme descrito na seção 16.6. É relativamente incomum configurar um servidor de nomes com a recursão desabilitada. Entretanto, pode ser que você queira permitir recursão para os seus próprios clientes, mas não para consultas externas.

A recursão pode ser controlada com uma precisão mais fina pela opção `allow-recursion` e uma lista de endereços que inclua os hosts e redes em cujo nome você pretende fazer consultas recursivas.

```
use-id-pool yes | no;          [no (somente V8)]
```

No BIND V8, essa opção faz com que **named** controle IDs de consulta pendentes para que ele não gere duplicatas e para que a seqüência de IDs que ele emite possa ser mais aleatória. Ela ajuda a impedir ataques de falsificação de DNS (DNS spoofing). Se você ativá-la, pode esperar por um emprego mais intenso de memória por parte de **named**. Entretanto, vale a pena o custo adicional com memória pelo recurso; recomendamos que você use essa opção. No BIND 9, a opção `use-id-pool` não existe mais e **named** sempre gerencia seus IDs de consulta desta maneira.

```
maintain-ixfr-base yes | no;          [no (somente V8)]
ixfr-base "nome-de-arquivo"           [(somente V8)]
ixfr-tmp-file "nome-de-arquivo"       [(somente V8)]
```

Transferências de zonas incrementais (consulte a RFC1995) permitem que os servidores enviem "remendos" (correções) para uma zona quando ela é modificada em vez de reenviar a zona inteira. Para uma zona como com, essa possibilidade de incremento é muito importante. A versão atual do BIND 8 permite transferências de zonas incre-

Snow666

mentais para qualquer zona que realize atualizações dinâmicas; ela mantém um arquivo log de transações se `maintain-ixfr-base` estiver configurado em `yes`. O BIND 9 sempre mantém um log.

No BIND 8, podemos interferir nos nomes-padrão para o log de transações IXFR e arquivo temporário pelas opções `ixfr-base` e `ixfr-tmp-file`. O tópico *Transferências de zonas*, na seção 16.12 aborda de forma mais detalhada as transferências de zonas incrementais.

`check-names { master|slave|response action }` *[veja o texto (somente V8)]*

O BIND 8 inclui código para verificar a validade de nomes de hosts — não no sentido de "Este host existe?", mas no sentido de "Este nome de host segue realmente as regras especificadas nas RFCs para nomes de hosts?". Um surpreendente número deles não o fazem.

Um nome é válido se ele contiver somente letras, números e travessões e não tiver um comprimento maior do que 64 caracteres por componente (incluindo o ponto) e 256 caracteres no total. As regras para nomes e a distinção entre a parte host e a parte domínio hoje são assuntos de debate. A internacionalização do sistema DNS e o suporte a conjuntos de caracteres não-ASCII poderão modificar todas as regras de atribuição de nomes vigentes atualmente.

Podemos especificar o comportamento de `check-names` de forma global e também individualmente para cada uma das zonas; uma especificação para uma zona prevalece em relação a valores globais. Podemos aplicar uma opção a servidores mestres, a servidores escravos ou a respostas retornadas de consultas.[9]

Cada tipo possui três valores possíveis para ação:

- `ignore` – não realiza nenhuma verificação
- `warn` – registra nomes incorretos em log, mas continua o processamento
- `fail` – registra em log e rejeita nomes incorretos

O padrão para o mestre é falhar, já que erros no mestre provavelmente estão na categoria de erros de digitação ou de nomes que não atendem às especificações que o administrador de sistemas deve corrigir. Um site não deve em sã consciência propagar nomes que não atendam às especificações. O padrão para servidores escravos é warn, e o padrão para respostas é ignore. Os valores-padrão são perfeitos; você não precisará alterá-los.

Dados de registro de `check-names` é um candidato em potencial para ajuste com o sistema de registro infinitamente flexível que é visto a partir da página 335.

O BIND 9, antecipando-se aos nomes de domínios internacionalizados, não implementa a opção `check-names`.

`transfer-format one-answer | many-answers;` *[veja o texto]*

Essa opção afeta a maneira pela qual registros de dados DNS (descritos na seção 16.11) são replicados por meio de servidores mestres para seus escravos. A transmissão em si de registros de dados costumava acontecer um registro de cada vez, que é uma receita para lerdeza e ineficiência. Uma opção para combinar vários registros num único pacote (`many-answers`) foi acrescentada no BIND 8.1; ela é o padrão no BIND 9. Use essa opção somente se todos os servidores com os quais você compartilhe dados de zonas estejam usando pelo menos a versão 8.1, já que os servidores BIND 4 não as compreende. Se você possuir um ambiente misto, poderá especificar um formato de transferência em instruções de servidor individuais para prevalecer em relação à opção global. Sua mescla de servidores determinará se você deve optar por `many-answers` globalmente e sobrepô-la para servidores específicos ou vice-versa.

`transfers-in` *número*;	[10]
`transfers-out` *número*;	[10 *(somente para V9)*]
`transfers-per-ns` *número*;	[2]
`transfer-source` *endereço-IP*;	[*dependente do sistema*]
`serial-queries` *número*;	[4 *(somente para V8)*]

Um site grande — um que atenda a uma zona muito ampla (como, por exemplo, com, que atualmente está além dos dois gigabytes) ou um que atenda a milhares de zonas — pode precisar de ajuste em algumas dessas opções de transferência de zonas.

As opções `transfers-in` e `transfers-out` limitam o número de transferências de zonas que chegam ou saem que podem acontecer concorrentemente.[10] A opção `transfers-per-ns` configura o número máximo de transferências de zonas que chegam que podem estar sendo realizadas concorrentemente por meio do mesmo servidor remoto.

9. Os servidores escravos no BIND 8 que usam `check-names` não parecem reconhecer nomes incorretos nas transferências de zonas e permitem que dados incorretos se propaguem. Entretanto, o recurso funciona bem para servidores mestres.
10. O código BIND força um limite definido internamente de 20 para o parâmetro `transfers-in`.

www.snow666.blogspot.com
Snow666

Sites grandes talvez precisem aumentar os parâmetros `transfers-in` ou `transfers-out`; tome cuidado para não ficar sem descritores de arquivos para o processo **named**. `transfers-per-ns` provavelmente não precisará ser alterado; ele controla quantos recursos o servidor mestre estará pretendendo dedicar a nós. Ele somente deve ser aumentado se todos os servidores mestres estiverem pretendendo manipular mais do que duas transferências de zonas simultâneas. Modificá-lo em uma base servidor por servidor através da estrutura `transfers` da instrução `server` é uma forma melhor de ajustar a convergência de zonas escravas.

A opção `transfer-source` permite que especifiquemos o endereço IP da interface que desejamos usar para transferências que chegam. Ela deve atender ao endereço especificado na instrução `allow-transfer` do mestre.

No BIND 8 podemos limitar o número de solicitações simultâneas para o número serial de uma zona por meio da opção `serial-queries`. Cada uma dessas solicitações mantém o estado no servidor local; se milhares de consultas estiverem sendo recebidas, esse limite poderá ajudar o servidor a manter sua sanidade mental. O valor-padrão é quatro, que está muito longe das expectativas de um site grande; eleve-o para várias centenas ou até mesmo para um milhar. No BIND 9 esse parâmetro é atualmente ignorado; ele será substituído por uma taxa de consultas no futuro.

Como acontece com qualquer parâmetro que modificamos drasticamente, devemos observar as coisas com atenção após modificar um desses valores reguladores para garantir que a máquina não apresente problemas.

Os arquivos de log são nossos aliados.

```
files número;                          [ilimitado]
```

A opção `files` configura o número máximo de arquivos que o servidor poderá ter aberto ao mesmo tempo. O valor-padrão, `unlimited (ilimitado)`, significa o mais próximo possível do número de arquivos abertos que o kernel é capaz de suportar. (Para modificar o número de arquivos aberto que o kernel pode suportar, configure o valor de **/proc/sys/fs/file-max**. Veja a seção 12.4.)

```
listen-on port endereço-IP lista-de-concordância-de-endereço;  [53 todos]
query-source address endereço-IP port porta-IP;                [aleatório]
```

A opção `listen-on` especifica as portas e interfaces de rede nas quais o processo **named** fica na escuta esperando consultas. A opção `query-source` especifica a interface e a porta que **named** usa para consultar outros servidores de nomes. Os valores dessas opções têm como padrão o comportamento-padrão de **named**: ouvir na porta 53 em todas as interfaces e usar uma porta UDP aleatória de numeração elevada e qualquer interface para consultas.

Veja a seção 22.4 para obter maiores informações sobre interfaces virtuais.

A opção `listen-on` nos permite executar vários servidores de nomes num único host. Poderíamos, por exemplo, querer proceder dessa maneira, pois é melhor não configurar um servidor BIND 4 ou BIND 8 para ser simultaneamente autorizado e para caching — estas versões de **named** preservam todos os seus dados num banco de dados gigantesco. **named** pode acabar ficando sem memória e os dados podem ser tornar corrompidos. Para evitar esse risco, podemos executar dois processos **named** separados: um como servidor autorizado e o outro como um servidor para caching que usa a instrução `listen-on` e um endereço IP virtual diferente. Os servidores autorizados e para caching poderão interagir como se estivessem em execução em máquinas distintas. Coloque somente o endereço IP do servidor de caching no arquivo **resolv.conf**. (Usar **ndc** para controlar várias instâncias de **named** rodando na mesma máquina pode se constituir num desafio ou mesmo uma tarefa quase impossível.)

Se seu site tiver um firewall, você poderá usar a opção `query-source` para atribuir a consultas DNS externas um perfil específico e reconhecível. Normalmente configuramos **named** para usar a porta 53 como a porta de origem, para que o firewall possa reconhecer tráfego DNS proveniente de fora como sendo pacotes confiáveis provenientes de um de nossos servidores de nomes. Não configure a porta de origem em zero — trata-se de uma porta ilegal, e **named** irá registrar em log a consulta como um erro e não responderá a ela. Um grande ISP possui um administrador de sistemas que gosta da porta 0 e tornou muitos de seus servidores de nomes inefetivos pelo uso da estrutura `query-source`. É curioso que seus clientes não percebam e não reclamem.

```
forwarders { in_addr; in_addr; … };            [empty list]
forward only | first;                          [first]
```

Em vez de fazer com que cada servidor de nomes execute suas próprias consultas externas, você pode designar um ou mais servidores como *encaminhadores*. Um servidor comum é capaz de pesquisar em seu cache e nos registros para os quais ele é autorizado; se ele não encontrar a resposta que está procurando, poderá então enviar a consulta para um host encaminhador. Dessa maneira, os encaminhadores constroem caches que beneficiam o site inteiro. Os encaminhadores reduzem a carga em sua rede, limitam o uso de CPU e de memória em servidores mais

Snow666

frágeis, aumentam o desempenho para os usuários e reduzam sua dependência na conectividade Internet externa. Muitos sites designam seus servidores mais poderosos e com mais capacidade de memória como encaminhadores.

Um site de porte médio pode construir um sistema DNS muito eficiente com uma série de servidores para caching que apontam para somente um ou dois encaminhadores. Sites grandes podem precisar de uma hierarquia de encaminhadores. O exemplo na página 302 usa um esquema de encaminhamento de dois níveis.

A opção `forwarders` lista os endereços IP dos servidores que você quer usar como encaminhadores. Eles são consultados alternadamente. O emprego de um encaminhador impede que o procedimento DNS normal inicie num servidor-raiz e siga a cadeia de referências. Tome cuidado para não criar loops de encaminhamento.

Um servidor somente para encaminhamento (`foward only`) coloca em cache valores e encaminhadores de consultas, mas ele jamais consulta alguém mais. Se os encaminhadores não responderem, as consultas falharão. Um servidor do tipo encaminhe primeiro (`forward first`) prefere lidar com encaminhadores, mas processará consultas diretamente caso seja necessário.

Já que a opção `forwarders` não possui nenhum valor-padrão, o encaminhamento não ocorre a menos que ele tenha sido configurado especificamente. Você pode ativar o encaminhamento de forma global ou então dentro das instruções de zonas individuais.

```
allow-query { lista-de-concordância-de-endereço };        [todos os hosts]
allow-transfer { lista-de-concordância-de-endereço };     [todos os hosts]
blackhole { lista-de-concordância-de-endereço };          [vazio]
```

Estas opções permitem que especifiquemos quais hosts (ou redes) podem consultar seu servidor de nomes e solicitar transferências de blocos de seus dados de zona. A lista de endereços `blackhole` identifica servidores com os quais você jamais irá querer falar; **named** não aceitará consultas desses servidores e não irá solicitar respostas deles.

```
sortlist { lista-de-concordância-de-endereço };           [deveria desaparecer, não a utilize]
```

Mencionamos a opção `sortlist` somente para alertá-lo para ficar longe dela. `sortlist` deve, na realidade, desaparecer. Seu propósito era o de ajudar os solucionadores primitivos que não ordenavam conjuntos de registros de forma apropriada. Ela permite que você especifique a ordem na qual várias respostas são retornadas e funciona contra a inteligência interna dos BINDs atuais.

Outras opções do BIND que se metem na ordem das coisas são a instrução `rrset-order`, que especifica se devem ser retornadas várias respostas numa ordem cíclica (round robin), fixa ou aleatória, e a instrução `topology`, que tenta adivinhar o sistema do BIND para seleção de servidores remotos para consulta. Na maioria dos casos não há nenhuma necessidade de usar essas instruções.

A instrução acl

Uma lista de controle de acesso é simplesmente uma lista de concordância de endereços com um nome:

```
acl nome-da-acl {
      lista-de-concordância-de-endereço
};
```

Podemos usar uma lista de controle de acesso em qualquer parte que uma lista de concordância de endereços seja plausível, com exceção da instrução `also-notify`.

Uma `acl` tem de ser uma instrução de primeiro nível em **named.conf** e, portanto, não tente colocá-la furtivamente no meio das demais declarações de opções. **named.conf** é lido de uma só vez, de modo que as listas de controle de acesso têm de ser definidas antes de serem usadas. São predefinidas quatro listas:

- `any` – todos os hosts
- `localnets` – todos os hosts na rede local
- `localhost` – a máquina em si
- `none` – nada

A lista `localnets` inclui todas as redes para as quais o host está conectado diretamente. Em outras palavras, ela é uma lista do módulo de endereços de rede da máquina em relação às suas máscaras de rede.

Snow666

A instrução server

named tem potencial para conversar com vários servidores, nem todos os quais estão rodando a versão mais recente de BIND e nem todos os quais nem mesmo são nominalmente sãos. A instrução `server` informa **named** sobre as características de seus parceiros remotos.

```
server endereço-IP {
   bogus yes | no;                          [no]
   provide-ixfr yes | no;                   [yes (somente V9)]
   request-ixfr yes | no;                   [yes (somente V9)]
   support-ixfr yes | no;                   [no (somente V8)]
   edns yes | no;                           [yes]
   transfers number;                        [2 (somente V9)]
   transfer-format one-answer | many-answers; [V8: one, V9: many]
   keys { ID-da-chave; ID-da-chave; … };
};
```

Você pode usar uma instrução `server` para anular os valores das opções de configuração relacionadas a servidor. Liste apenas aquelas para as quais você quer um comportamento não padronizado.

Se você marcar um servidor como `bogus`, **named** não enviará nenhuma consulta em sua direção. Essa diretiva deve em geral ser reservada para servidores que realmente são não-genuínos.

As cláusulas `ixfr` mudaram entre as versões 8 e 9, embora ambos os conjuntos sejam similares. V8 possui `support-ixfr` e V9 possui `provide-ixfr` e `request-ixfr`. As instruções `server` no arquivo de configuração de um servidor rodando a V8 pode configurar `support-ixfr` em `yes` se o servidor remoto compreender transferências de zonas incrementais. Um servidor V9 atuando como mestre para uma zona fará transferências de zonas incrementais se `provide-ixfr` estiver configurado em `yes`. Da mesma forma, um servidor rodando a V9 atuando como escravo solicitará transferências de zonas incrementais do mestre caso `request-ixfr` esteja configurado em `yes`.

A cláusula `edns` determina se o servidor local tentará usar o protocolo DNS estendido ao contactar o servidor remoto. Muitos dos recursos novos do BIND (IPv6 e DNSSEC, por exemplo) geram pacotes maiores do que 512 bytes e, portanto, requerem o emprego do protocolo EDNS para negociar um tamanho de pacote UDP maior.

A cláusula `transfers` limita o número de transferências de zonas de fora para dentro concorrentes do servidor remoto. Ela é uma versão específica por servidor da instrução `transfers-in`, mas como se aplica somente a um servidor, ela atua como uma sobreposição da opção `transfers-per-ns` por servidor.

O nome é diferente para preservar compatibilidade com o BIND 8.

As cláusulas `transfer-format` são as formas específicas a servidor das opções discutidas na página 289. Use `transfer-format` caso você converse tanto com servidores BIND 8/9 quanto servidores BIND 4 ou se você tiver servidores de nomes Microsoft em sua rede. O NT não é capaz de manipular nada além do formato original de uma resposta; o Windows 2000 se adapta a qualquer um dos formatos.

A cláusula `keys` identifica um ID chave que foi previamente definido numa instrução `key` para emprego com assinaturas de transação TSIG (veja a seção 16.13). Quaisquer solicitações enviadas para o servidor remoto são assinadas com essa chave. Solicitações originadas no servidor remoto não precisam ser assinadas, mas se forem, a assinatura será verificada.

A instrução logging

named é o atual detentor do "sistema de registro mais configurável da face da Terra". Syslog coloca a priorização das mensagens de log nas mãos do programador e a administração dessas mensagens, nas mãos do administrador de sistemas. Mas para uma dada prioridade, o administrador de sistemas não tinha nenhuma maneira de dizer: "Eu me importo com esta mensagem mas não com aquela". O BIND 8 adicionou categorias que classificam mensagens de log por tipo e canais que alargam as opções para a administração das mensagens. As categorias são determinadas pelo programador e os canais pelo administrador de sistemas.

Já que a questão de registro é um tanto tangencial (especialmente dada a quantidade de explicação exigida), a discutimos no tópico sobre depuração neste capítulo, na seção 16.14.

A instrução zone

As instruções `zone` são o cerne do arquivo **named.conf**. Elas informam **named** sobre as zonas para as quais ele é autorizado e configura as opções que são apropriadas para gerenciar cada uma das zonas. Uma instrução `zone` também é usada para pré-carregar as dicas do servidor-raiz (os nomes e os endereços dos servidores-raiz, que disparam o processo de pesquisa DNS).

www.snow666.blogspot.com
Snow666

O formato exato de uma instrução `zone` varia, dependendo do papel que **named** desempenhará em relação àquela zona (por exemplo, servidor mestre ou servidor escravo). Examinamos uma possibilidade por vez. Muitas das opções globais abordadas anteriormente podem se tornar parte de uma instrução `zone` e prevalecerem sobre os valores definidos previamente. Não repetiremos essas opções aqui, exceto algumas usadas mais freqüentemente.

Configurando o servidor mestre para uma zona
Eis o formato de que você precisa para uma zona da qual este processo **named** é o servidor mestre:

```
zone "nome-de-domínio" {
   type master;
   file "caminho";
};
```

Outros atributos específicos a determinado servidor também são especificados com freqüência. Por exemplo:

```
allow-query { lista-de-concordância-de-endereço };        [todos]
allow-transfer { lista-de-concordância-de-endereço };     [todos]
allow-update { lista-de-concordância-de-endereço };       [nenhum]
ixfr-base "caminho";                                      [nome-de-domínio.ixfr (somente V8)]
zone-statistics yes | no                                  [no]
```

`nome-de-domínio` numa especificação de zona sempre tem de estar entre aspas duplas.

Os dados da zona são mantidos em disco num arquivo de formato legível (e editável). Já que não há nenhum padrão para o nome de arquivo, você tem de fornecer uma instrução `file` ao declarar uma zona mestre. Um arquivo de zonas é simplesmente um conjunto de registros de recursos DNS; a descrição de formato começa na seção 16.11.

As opções de controle de acesso não são exigidas, porém é uma boa idéia usá-las. Se forem usadas atualizações dinâmicas para essa zona, a cláusula `allow-update` tem que estar presente com uma lista de concordância de endereços que limita os hosts com base nas quais as atualizações podem ocorrer. As atualizações dinâmicas se aplicam somente a zonas mestres; a cláusula `allow-update` não pode ser usada para uma zona escrava (no BIND 9). Certifique-se de que essa cláusula inclua somente seus servidores DHCP locais e não toda a Internet.[11]

Se tiverem de ser usadas transferências de zonas incrementais com essa zona, o BIND 8 mantém um log de transações num arquivo chamado *nome-de-domínio*.**ixfr** no diretório-raiz de **named**. Se você quiser alterar o nome, use a cláusula `ixfr-base` para especificar um novo nome de arquivo. Esse arquivo é mantido por **named** e não precisa de nenhuma atenção de sua parte.

No BIND 9, o log de transações é usado tanto para atualizações dinâmicas como para IXFR. Seu nome termina com **.jnl** e não é configurável. Tanto as atualizações dinâmicas quanto as transferências de zonas incrementais são recursos relativamente novos do BIND. Eles são vistos com mais detalhes na seção 16.12.

A opção `zone-statistics` faz com que **named** controle as informações estatísticas de consultas/respostas como o número e porcentagem de respostas que eram referências, recursão demandada ou que resultaram em erros. Veja os exemplos para o BIND 8 e BIND 9 na seção 16.14.

Com todas essas opções específicas a uma determinada zona (e cerca de 30 outras que não apresentamos aqui), a configuração está começando a ficar complicada. Entretanto, uma declaração de zonas mestres formada por nada além de um nome de caminho para o arquivo de zonas é perfeitamente razoável. No BIND 4, isso é tudo o que você poderia especificar. Eis um exemplo da documentação BIND que modificamos ligeiramente:

```
zone "example.com" {
   type master;
   file "forward/example.com";
   allow-query { any; };
   allow-transfer { my-slaves; };
}
```

`my-slaves` é uma lista de controle de acesso que foi definida previamente.

Configurando um servidor escravo para uma zona
A instrução `zone` para um servidor escravo é muito similar àquela de um servidor mestre:

```
zone "nome-de-domínio" {
   type slave | stub;
   file "caminho";
```

11. Você também precisa de filtragem de ingresso no seu firewall; veja a seção 21.10. Melhor ainda, use o TSIG para autenticação.

Snow666


```
   ixfr-base "caminho";                                    [somente V8]
   masters { endereço-IP; endereço-IP; … };                [nenhum padrão]
   allow-query { lista-de-concordância-de-endereço };      [all]
   allow-transfer { lista-de-concordância-de-endereço };   [all]
};
```

Servidores escravos normalmente mantêm uma cópia completa do banco de dados de suas zonas. Porém, se `type` for configurado para `stub` em vez de um escravo, somente registros NS (Name Server) são transferidos. Zonas stub permitem que os processos **named** para a zona-pai descubram automaticamente que máquinas fornecem serviços DNS para suas zona-filha delegadas, na eventualidade do administrador da zona-filha não estar ciente sobre a necessidade de informar a zona-pai sobre as alterações. A zona-pai precisa dessas informações para fazer referências apropriadas ou consultas recursivas. Revisaremos esse tópico com mais detalhes na seção 16.11.

A instrução `file` especifica um arquivo local no qual o banco de dados replicado pode ser armazenado. Cada vez que o servidor busca uma nova cópia da zona, ele salva os dados nesse arquivo. Se o servidor travar e reinicializar, o arquivo pode então ser recarregado no disco local sem ser transferido pela rede.

Você não deve editar este arquivo cache, já que ele é mantido por **named**. Entretanto, pode ser interessante dar uma olhada nele caso você suspeite que tenha cometido um erro no arquivo de dados do servidor mestre. O arquivo de disco do escravo nos mostra como **named** interpretou os dados de zona originais; nomes relativos e diretivas de origem foram expandidos. Se você vir um nome no arquivo de dados que se pareça como um destes a seguir,

```
128.138.243.151.cs.colorado.edu.
anchor.cs.colorado.edu.cs.colorado.edu.
```

pode estar certo de que acabou esquecendo algum ponto final em algum lugar.

A instrução `masters` lista os endereços IP de uma ou mais máquinas das quais o banco de dados de zonas pode ser obtido. Dissemos que somente uma máquina pode ser o mestre para uma zona e, assim, por que é possível listar mais do que um endereço? Duas razões.

Em primeiro lugar, a máquina mestra poderia ter mais de uma interface de rede e, conseqüentemente, mais de um endereço IP. É possível para uma interface se tornar inatingível (devido a problemas de rede ou de roteamento) enquanto outras ainda continuem acessíveis. Portanto, é uma boa prática listar todos os endereços topologicamente distintos do servidor mestre.

Em segundo lugar, **named** realmente não se importa de onde provêm os dados. Ele pode extrair igualmente o banco de dados de um servidor escravo como de um servidor mestre. Você poderia usar esse recurso para permitir que um servidor escravo bem conectado servisse como uma espécie de mestre-backup, já que os endereços IP seriam tentados em ordem até que um servidor operacional fosse encontrado. Teoricamente, você também pode configurar a hierarquia de servidores, com um mestre atendendo a vários servidores de segundo nível que, por sua vez, servem vários servidores de terceiro nível.

Sugerimos que você liste somente os endereços de servidor mestre confiáveis na linha `masters`.

Configurando as dicas-raiz

Uma outra forma de instrução `zone` aponta **named** em direção a um arquivo por meio do qual ela pode munir (pré-carregar) seu cache com os nomes e endereços dos servidores de nomes raiz:

```
zone "." {
   type hint;
   file "caminho";
};
```

As "dicas" são um conjunto de registros DNS que listam servidores para o domínio-raiz (".") Elas são necessárias para dar a **named** um ponto de partida para busca de informações sobre os domínios dos demais sites. Sem elas, **named** saberia apenas sobre os domínios que ele efetivamente atende e seus subdomínios.

O arquivo de dicas é freqüentemente chamado de **root.cache**; ele contém a resposta que você obteria caso consultasse um servidor raiz em busca de registros de servidor de nomes no domínio ".". Discutiremos como configurar um arquivo de dicas apropriado a partir da página 334.

O BIND 9 possui dicas de servidor raiz compiladas em seu código, de modo que nenhuma configuração da zona-raiz é realmente necessária. Entretanto, se você fornecer um arquivo de dicas, o BIND 9 o usará. Recomendamos que você forneça dicas explícitas; a política invadiu a arena do DNS, tornando os servidores de nomes raiz e seus endereços IP mais voláteis.

Snow666

Configurando uma zona de encaminhamento
Uma zona do tipo `forward` prevalece em relação às configurações de encaminhamento global de **named** (descritos na página 290) para um determinado domínio:

```
zone "nome-de-dominio" {
   type forward;
   forward only | first;
   forwarders { endereço-IP; endereço-IP; … };
};
```

Você poderia usar uma zona de encaminhamento caso sua organização tivesse um acordo estratégico em vigência com algum outro grupo ou empresa e você quisesse afunilar o tráfego diretamente para os servidores de nomes daquela empresa, evitando o caminho de consultas-padrão. Você poderia usar tal arranjo para acessar servidores de nomes que eram invisíveis ao mundo externo, como é feito no exemplo da seção 16.10.

A instrução key

A instrução `key` define uma chave criptográfica para ser usada para autenticação com determinado servidor. São dadas informações de background sobre o suporte de BIND à autenticação criptográfica na seção 16.13, *Questões de Segurança*. Aqui somente vemos superficialmente a mecânica do processo.

Para criarmos um registro-chave, especificamos tanto o algoritmo criptográfico que desejamos usar como um "segredo compartilhado", representado na forma de uma string codificada base 64:

```
key ID-da-chave {
   algorithm string;
   secret string;
};
```

Como acontece com as listas de controle de acesso, *ID-da-chave* tem de ser definido antes de ser usado. Para associar a chave a um servidor, basta incluir *ID-da-chave* na cláusula `keys` da instrução `server` desse servidor. A chave é usada tanto para verificar solicitações desse servidor quanto para assinar as respostas a essas solicitações.

A instrução trusted-keys

A instrução `trusted-keys` destina-se à segurança DNSSEC, especificada na RFC2065. Cada entrada é um conjunto de 5 valores que identifica o nome de domínio, flags, protocolo, algoritmo e chave necessários para conversar de forma segura com um servidor de nomes para esse domínio. O formato é:

```
trusted-keys {
   domínio flags protocolo algoritmo chave;
   domínio flags protocolo algoritmo chave;
   …
}
```

Cada linha representa a chave confiável para determinado domínio. *flags*, *protocolo* e *algoritmo* são inteiros não negativos. *chave* é uma string codificada base 64.

A construção `trusted-keys` é destinada a ser usada quando uma zona tem assinatura, mas sua zona-pai não, de modo que você não pode ter certeza de que a chave pública para a zona que obtém do DNS seja realmente autêntica. Introduzir uma chave confiável por meio de uma instrução `trusted-keys` (usando métodos desvinculados) garante que você realmente esteja de posse da chave apropriada para o domínio em questão.

DNSSEC é visto com mais detalhes a partir da seção 16.13.

A instrução controls

A instrução `controls` especifica como **ndc** (**rndc** no BIND 9) controla um processo **named** em execução. **ndc** é capaz de iniciar e interromper **named**, fazer um dump de seu estado, colocá-lo no modo de depuração etc. **ndc** é um programa de rede e sem uma configuração apropriada pode permitir que qualquer um na Internet interfira no seu servidor de nomes. A sintaxe é a seguinte:

```
BIND 8: controls {
   inet endereço-IP port Número-porta allow { lista-de-concordância-de-endereço | chave ... };
   unix permissão proprietário grupo;                                          [0600 0 0]
}
```

Snow666


```
BIND 9: controls {
   inet endereço-IP allow { lista-de-concordância-de-endereço } keys { lista-de-chaves }; [veja o texto]
}
```

O BIND 9 também permite que a porta seja especificada, porém o padrão é a porta 953 caso a cláusula `port` não esteja presente. Se a linha `unix` estiver presente, o BIND 9 a ignora.

Permitir que o seu servidor de nomes seja controlado remotamente parece algo prático, porém, ao mesmo tempo, perigoso. Autenticação rigorosa através de uma entrada `key` na cláusula `allow` era opcional no BIND 8. O BIND 9 requer autenticação rigorosa, ignora chaves da lista de concordância de endereços e usa as chaves explicitamente declaradas na cláusula `keys` da instrução `controls`.

No BIND 9, você pode usar o comando **rndc-confgen** para gerar uma chave de autenticação para emprego entre **rndc** e **named**. Há essencialmente duas maneiras de configurar isso: fazer com que tanto **named** quanto **rndc** consultem o mesmo arquivo de configuração para tomar conhecimento da chave (**/etc/rndc.key**) ou então você poderá informar cada programa da chave em seu próprio arquivo de configuração (**/etc/rndc.conf** para **rndc** e **/etc/named.conf** para **named**). A última opção é mais complicada, porém ela é necessária quando **named** e **rndc** rodam em computadores diferentes.

Quando nenhuma instrução `controls` estiver presente, o BIND 9 usa como padrão o endereço de loopback para a lista de concordância de endereços e procura a chave em **/etc/rndc.key**. Pelo fato de a autenticação rigorosa ser obrigatória no BIND 9, você não pode usar o comando **rndc** para controlar **named** caso não haja nenhuma chave. Isso pode parecer um tanto draconiano, mas considere o seguinte: mesmo se **rndc** funcionasse somente no host local (127.0.0.1) e este endereço fosse bloqueado do mundo externo no seu firewall, você ainda estaria confiando que todos os usuários locais não bagunçariam seu servidor de nomes. Qualquer usuário poderia enviar um **telnet** para a porta de controle e digitar "stop" — um ataque de recusa de serviços bastante efetivo.

Eis um exemplo da saída (para padronizar) gerada por **rndc-confgen** quando uma chave de 256 bits é solicitada (optamos por 256, pois ela se encaixa na página!). Normalmente você redirecionaria a saída para **/etc/rndc.conf**. Os comentários na parte inferior da saída mostram as linhas que precisam ser adicionadas a **named.conf** para fazer com que **named** e **rndc** funcionem juntos.

```
% ./rndc-confgen -b 256
# Início de rndc.conf
key "rndc-key" {
   algorithm hmac-md5;
   secret "orZuz5amkUnEp52zlHxD6cd5hACldOGsG/elP/dv2IY=";
};

options {
   default-key "rndc-key";
   default-server 127.0.0.1;
   default-port 953;
};
# Final de rndc.conf

# Use o seguinte em named.conf, ajustando a lista de permissão conforme necessário:
# key "rndc-key" {
#     algorithm hmac-md5;
#     secret "orZuz5amkUnEp52zlHxD6cd5hACldOGsG/elP/dv2IY=";
# };
#
# controls {
#     inet 127.0.0.1 port 953
#        allow { 127.0.0.1; } keys { "rndc-key"; };
# };
# Final do arquivo named.conf
```

O modo "automático", usado para gerar um arquivo de configuração compartilhado, é mais simples. Executar como root, **rndc-confgen -a -b 256** produz o arquivo **/etc/rndc.key**, que contém

```
key "rndc-key" {
      algorithm hmac-md5;
      secret "laGbZj2CobycOm/jFVNCu8OJzsLKNH+CCb2JCWY6yJw=";
};
```

Snow666

O arquivo possui modo 600 e seu proprietário deve ser o usuário **named** ou o root.

No BIND 8, **ndc** também pode contactar **named** através de um socket de domínio UNIX chamado **/var/run/ndc**. A linha de configuração `unix` configura as permissões e propriedades para esse socket e, conseqüentemente, restringe o acesso a ele. O parâmetro *permissão* deve ser um número octal que representa o modo do socket; os parâmetros *proprietário* e *grupo* são o UID e GID do proprietário do socket. Os padrões são permissão de leitura e de escrita somente para root.

Consulte a seção 13.4 para obter maiores informações sobre espaços de endereços privados.

DNS subdividido e a instrução view do BIND 9

Muitas instalações querem que a visão interna de suas redes seja diferente daquela vista externamente por quem as acessa via Internet. Esse tipo de configuração (algumas vezes denominado "DNS subdividido") parece ser cada vez mais popular.

Por exemplo, você poderia revelar todos os hosts de uma zona a usuários internos, mas restringir a visão externa a alguns poucos servidores bem conhecidos. Ou então, você poderia expor o mesmo conjunto de hosts em ambas as vistas, mas fornecer registros adicionais (ou diferentes) para usuários internos. Por exemplo, os registros MX para encaminhamento de e-mail poderiam apontar para uma única máquina centralizadora de correio (e-mail) de fora do domínio, mas apontar para estações de trabalho individuais sob a perspectiva de usuários internos.

Uma configuração DNS subdividida é especialmente útil para instalações que usam espaços de endereços IP privados baseados na RFC1918 em suas redes internas. Por exemplo, uma consulta buscando o nome de host associado ao endereço IP 10.0.0.1 jamais poderá ser respondida pelo sistema DNS global, mas faz sentido dentro do contexto da rede local. Das consultas que chegam aos servidores de nomes raiz, 4% a 5% são *de* um endereço IP em um dos intervalos de endereços privados ou *sobre* um destes endereços. Nenhuma delas pode ser respondida; ambas são resultantes de uma configuração incorreta, seja do DNS subdividido do BIND ou por parte dos "domínios" Microsoft.

Para criar uma configuração DNS subdividida com o BIND 8, você configura servidores distintos para as versões interna e externa da realidade. Os nomes são divididos em zonas internas e externas. Clientes locais apontam para servidores que liberam a versão interna da zona; os registros NS da zona-pai apontam para servidores que armazenam a versão externa. Transferências e consultas recursivas são limitadas. A Tabela 16.8 ilustra as diferenças de configuração entre os servidores de nomes interno e externo.

Tabela 16.8 Configurações de servidor interna e externa para DNS subdividido no BIND 8.

Parâmetro	Externa	Interna
`directory`	`/var/named/external`	`/var/named/internal`
`listen-on`	*interface externa*	*interface interna*
`recursion`	`não`	`sim`
`allow-query`	`qualquer`	*somente redes internas*
`forward`	*não utilizado*	`somente encaminhamento`
`forwarders`		`servidores de nomes externos`

A instrução `view` do BIND 9 simplifica a definição das configurações do DNS subdividido colocando ambos os conjuntos de dados dentro da mesma cópia de **named**. A instrução `view` empacota uma lista de acesso que controla quais clientes vêem quais vistas, algumas opções que se aplicam a todas as zonas da vista e, finalmente, as zonas em si. A sintaxe é:

```
view nome-da-vista {
   match-clients { lista-de-concordância-de-endereço } ;
   opção-da-vista; …
   instrução-zone; …
} ;
```

A cláusula `match-clients` controla quem pode ver a vista. As vistas são processadas em ordem; portanto, coloque as vistas mais restritas em primeiro lugar. As zonas nas diversas vistas podem ter os mesmo nomes. As vistas são uma proposição ou tudo ou nada; se você usá-las, todas as instruções `zone` contidas em seu arquivo **named.conf** deverão aparecer no contexto de uma vista.

Eis um exemplo extraído da documentação do BIND 9 que simula o esquema DNS subdividido, descrito anteriormente. As duas vistas definem a mesma zona, porém com dados diferentes.

Snow666


```
view "internal" {
   match-clients { our_nets; };          // somente redes internas
   recursion yes;                        // somemte clientes internos
   zone "example.com" {                  // vista completa da zona
      type master;
      file "example-internal.db";
   };
};
view "external" {
   match-clients { any; };               // permite todas as consultas
   recursion no;                         // mas não recursão
   zone "example.com" {                  // somente hosts "públicos"
      type master;
      file "example-external.db";
   }
};
```

Se a ordem das vistas fosse invertida, ninguém jamais veria a vista interna. Os hosts internos concordariam com o valor `any` na cláusula `match-clients` da vista externa antes de atingirem a vista interna.

Nosso segundo exemplo de configuração DNS (iniciando na seção 16.10) usa DNS subdividido para reforçar um único gateway de e-mails que chegam via registros MX, para ocultar várias sub-redes dos endereços privados e tornar inacessíveis por parte do mundo externo os registros HINFO e TXT que contêm informações confidenciais.

16.10 Exemplos de Configuração BIND

Agora que exploramos as maravilhas de **named.conf**, vamos dar uma olhada em alguns exemplos completos. Nas seções a seguir, ilustramos três modelos de configuração:

- Um computador Linux doméstico de um estudante
- Uma pequena empresa especializada em segurança de sistemas que usa DNS subdividido
- Um departamento de uma universidade que usa uma hierarquia de encaminhamento de três níveis

Um computador doméstico com Linux

Rob Braun, um estudante, possui um computador com Linux em sua casa que fornece serviço de nomes DNS para seu domínio, synack.net, e para os domínios de uma série de amigos. Ele está usando BIND 8.2.3. Seu arquivo **named.conf** é mostrado a seguir. Adicionamos alguns comentários ao final de algumas linhas.

A configuração de Rob é bastante simples. As opções são em sua maioria os padrões: um servidor recursivo, arquivos nos lugares usuais, consultas e transferências OK, um formato de transferência com uma resposta, dialogando na porta 53 etc. É exercido algum controle sobre a configuração de zonas individuais. Por exemplo, as zonas synack.net e xinetd.org permitem transferências somente para um host e elas não permitem atualizações dinâmicas.

Este servidor de nomes é o servidor-mestre para dois domínios: synack.net e xinetd.org. Ele também é um servidor escravo para os domínios de dois amigos: teich.net e rmtai.com. Como seria de esperar na configuração de um estudante de computação, a seção de registro não é o padrão. Ela inclui depuração indulgente e canais de curiosidade modelados em cima do arquivo de configuração que vem com o BIND.

```
/* arquivo named.conf, gw.synack.net */

options {
   directory "/var/named";
   pid-file "/var/named/named.pid";
};

zone "synack.net" {
   type master;
   file "synack.forw";
   allow-transfer { 198.11.19.15; };
};

zone "xinetd.org" {
   type master;
```

Snow666


```
   file "xinetd.forw";
   allow-transfer { 198.11.19.15; };
};

zone "1.168.192.in-addr.arpa" {
   type master;                    // reverso, para endereços privados
   file "named.rev";
};

zone "." {
   type hint;
   file "cache.db";
};

zone "teich.net" {
   type slave;
   file "teich.net.sec";
   masters { 216.103.220.218; };
};

zone "rmtai.com" {
   type slave;
   file "rmtai.com.sec";
   masters { 216.103.220.218; };
};

// Define três canais de registro (mensagens de syslog importantes,
// depuração moderada e mensagens de zonas de carregamento) e
// depois associa categorias a eles.

logging {
   channel syslog_errors {
      syslog local1;
      severity error;
   };
   channel moderate_debug {
      severity debug 3;     // depuração nível 3
      file "foo";           // para o arquivo foo
      print-time yes;       // exibe o horário das entradas de log
      print-category yes;   // imprime o nome da categoria
      print-severity yes;   // imprime o nível de gravidade
   };
   channel no_info_messages {
      syslog local2;
      severity notice;
   };
   category parser {
      syslog_errors;
      default_syslog;
   };
   category lame-servers { null; };      // não registra estas
   category load { no_info_messages; };
   category default {
      default_syslog;
      moderate_debug;
   };
}; // final da instrução  logging
```

Essa configuração não possui nenhuma zona localhost reversa; o host local tem de ser associado ao arquivo **/etc/hosts**.

Snow666

Uma pequena empresa especializada em segurança de sistemas

Nosso próximo exemplo é o de uma pequena empresa especializada em consultoria sobre segurança. Eles usam BIND 9 no Red Hat 7.2 e usam vistas para implementar um sistema DNS subdividido no qual usuários internos e externos vêem dados de host diferentes. Eles também usam espaço de endereços privado internamente; consultas sobre esses endereços jamais devem escapar para a Internet para bagunçar o sistema DNS global. Eis o arquivo **named.conf**, reformatado e com alguns comentários:

```
options {
   directory "/var/domain";
};

controls {
   inet 127.0.0.1 allow { 127.0.0.1; } keys { atkey; };
};

include "atrust.key";                        // definição de atkey no arquivo modo 600

view "atrust" {                              // vista interna

   match-clients { 192.168.1.0/24; 192.168.2.0/24; };
   recursion yes;

   zone "." {                                // zona de dicas raiz
      type hint;
      file "named.ca";
   };
   zone "localhost" {                        // zona localhost direta
      type master;
      file "localhost.zone";
      allow-update { none; };
   };
   zone "0.0.127.in-addr.arpa" {             // zona localhost reversa
      type master;
      file "named.local";
      allow-update { none; };
   };
   zone "atrust.com" {                       // zona direta interna
      type master;
      file "atrust.com-internal";
   };
   zone "2.168.192.in-addr.arpa" {           // zona reversa interna
      type master;
      file "192.168.2.rev";
      allow-update { none; };
   };
   zone "1.168.192.in-addr.arpa" {           // zona reversa interna
      type master;
      file "192.168.1.rev";
      allow-update { none; };
   };
   zone "kalos-strategy.com" IN {            // zona para a qual somos secundários
      type forward;
      forwarders { 127.0.0.1; };
   };
};                                           // final da vista interna
view "world" {                               // vista externa

   match-clients { any; };
   recursion no;

   // instruções de zona para "." (ponto) e localhost como acima, omitido por brevidade
```

Snow666


```
   zone "atrust.com" {                              // zona  externa direta
      type master;
      file "atrust.com-external";
   };

   zone "kalos-strategy.com" IN {                   // zona para a qual somos secundários
      type slave;
      file "kalos-strategy.com";
      masters { 66.1.239.14; };
   };
   zone "198.168.206.in-addr.arpa" {                // zona revesra externa
      type master;
      file "206.168.198.rev";
      allow-update { none; };
   };
};                                                  // final da vista externa
```

O arquivo **atrust.key** contém a definição da chave chamada "atkey":

```
key "atkey" {
    algorithm hmac-md5;
    secret "segredo compartilhado entra aqui";
};
```

São necessárias duas entradas para kalos-strategy.com, uma empresa para qual atrust.com executa um servidor escravo. Se a zona fosse mencionada somente na vista externa, as consultas provenientes de fora do domínio atrust.com seriam aceitáveis, porém consultas provenientes de dentro sairiam pela rede para fazer uma pergunta para a qual o próprio servidor de nomes de atrust.com é autorizado. Uma solução para essa ineficiência é mencionar kalos-strategy.com na vista interna, porém listá-lo como uma zona direta para que o servidor de nomes veja a consulta com sua função externa também.

Os arquivos de zonas **atrust.com-internal** e **atrust.com-external** definem as vistas separadas. Trechos de código deles são mostrados a seguir. Em primeiro lugar, o arquivo interno:

```
; atrust.com-internal file
$TTL 86400
$ORIGIN com.
atrust              3600   SOA  ns1.atrust.com. trent.atrust.com. (
                                     2001110500 10800 1200 3600000 3600 )
                    3600   NS   NS1.toadranch.com.
                    3600   NS   NS2.toadranch.com.
                    3600   MX   10 mailserver.atrust.com.
                    3600   A    206.168.198.209
$ORIGIN atrust.com.
www                        A    192.168.2.1
mailserver                 A    192.168.2.1
bull                       A    192.168.1.1
superg                     A    192.168.1.249
airport                    A    192.168.1.250
superg                     A    192.168.1.252
at-dmz-gw                  A    192.168.1.254
bark                       A    192.168.2.1
at-dmz-outside-gw          A    192.168.2.253
at-external-gw             A    192.168.2.254
at-external-outside-gw     A    206.168.198.220
indras-gw                  A    206.168.198.222
```

E eis a vista externa deste mesmo domínio, atrust.com:

```
; atrust.com-external file
$TTL 86400
$ORIGIN com.
atrust                          3600   SOA  ns1.atrust.com. trent.atrust.com. (
                                                2001110601 10800 1200 3600000 3600 )
                                3600   NS   NS1.toadranch.com.
```

Snow666


```
                                3600  NS    NS2.toadranch.com.
                                3600  MX    10 mailserver.atrust.com.
                                3600  A     206.168.198.209

$ORIGIN atrust.com.
www                                 A       206.168.198.209
mailserver                          A       206.168.198.209
ns1                                 A       206.168.198.209
bark                                CNAME   www

; mapas reversos                  ### delegados com CNAME a partir de
;                                 ### servidores 168.198.209.in-addr.arpa

exterior1                         IN A      206.168.198.209
209.198.168.206                   IN PTR    exterior1.atrust.com.
exterior2                         IN A      206.168.198.213
213.198.168.206                   IN PTR    exterior2.atrust.com.
exterior3                         IN A      206.168.198.220
220.198.168.206                   IN PTR    exterior3.atrust.com.
```

Note que quando são necessários nomes múltiplos para a mesma máquina, eles são apresentados como registros A adicionais em vez de registros CNAME. Um cliente pesquisando o nome www.atrust.com provocaria uma pesquisa extra se www fosse associado ao servidor em si através de um CNAME e, portanto, potencialmente dobrando o tempo que ele levaria para pesquisar o endereço IP.

O TTL nesses arquivos de zonas é configurado como padrão para um dia (86.400 segundos), mas é reiniciado para uma hora (3.600 segundos) para os registros SOA e NS. A maioria dos registros individuais nos arquivos de zonas não recebe um valor TTL explícito. O TTL é opcional; ele pode ser adicionado a qualquer linha individual no arquivo de zonas logo antes do tipo de registro.

Os bizarros registros PTR no final do arquivo externo permitem o ISP de atrust.com delegar a associação reversa de um trecho muito pequeno do espaço de endereços. Os registros CNAME no site do ISP fazem com que essa variação do uso de CNAME funcione; veja a seção *O truque CNAME* para a maiores informações.

Um departamento de uma universidade

O departamento de ciências de computação na Universidade do Colorado executa o BIND 8 numa cela onde é aplicado **chroot** e implementa DNS subdividido executando duas instâncias de **named** na mesma máquina. A zona interna contém todos os hosts locais, inclusive alguns locados no espaço de endereços privados. A zona externa inclui a maioria dos hosts, mas possui registros MX apontando para o centralizador de e-mail visível externamente em vez de para cada máquina individual. Obviamente, a vista externa não lista nenhuma máquina que reside no espaço de endereços privado. Eis os comandos usados para iniciar as duas instâncias:

```
/usr/local/sbin/named -t /var/named -u named -c /etc/named-internal.conf
/usr/local/sbin/named -t /var/named -u named -c /etc/named-external.conf
```

Os arquivos de configuração não se chamam realmente **/etc/named-*.conf** conforme mostrado, mas sim **/var/named/etc/named-*.conf**, já que o processo **chroot** é aplicado ao diretório **/var/named**. Internamente, Colorado usa servidores para caching em cada sub-rede; os servidores para caching encaminham para uma lista de servidores escravos. Os escravos encaminham para um servidor mestre que contacta a Internet em seu nome. Cada servidor de encaminhamento é configurado como *forward first*. A seguir mostramos as três configurações: os servidores somente para caching, os servidores escravo e o mestre. Todos os servidores rodam uma versão atualizada do BIND 8.

A configuração somente para caching é apropriada para uma sub-rede que quer um servidor local, mas não quer sujar suas mãos com arquivos de dados. Precisamos somente configurar o arquivo **named.conf** e o arquivo de dicas e iniciar **named** no momento do boot. Não há realmente zonas locais no arquivo de zonas, somente as dicas de servidor-raiz e a zona localhost reversa. Os servidores somente para caching fazem parte da vista interna e não rodam num ambiente em que foi aplicado **chroot**.

```
// arquivo de configuração para o bind 8.2 – servidor para caching

// Opções globais
options {
```

Snow666


```
   directory "/var/named";
   named-xfer "/usr/local/sbin/named-xfer"; // somente para bind 8
   // constrói um rico cache em nosso mestre e escravos oficiais
   forwarders {
      128.138.243.151;   // mroe
      128.138.243.140;   // anchor
      128.138.243.137;   // moet
      128.138.243.138;   // vulture
      128.138.236.20;    // piper
   };
   forward first;
   query-source address * port 53;
};

// Registro, syslog para local3, nenhum servidor defeituoso
logging {
   channel syslog_info {
      syslog local3;
      severity info;
   };
   category lame-servers { null; };
   category default { syslog_info; };
};

zone "." { // Cache de servidores-raiz
   type hint;
   file "named.cache";
};

//Servidores mestres para a zona localhost reversa
zone "0.0.127.in-addr.arpa" {
   type master;
   file "localhost";
   notify no;
};
```

O arquivo de configuração para servidores escravos inclui a zona direta cs.colorado.edu e várias zonas reversas que reduzimos para uma ou duas para fins de ilustração. Neste exemplo, as zonas reversas não são subdivididas num limite de byte (elas são na maioria /26s), mas como todas as quatro sub-redes são controladas pela mesma autoridade administrativa, elas são mantidas no mesmo arquivo e a utilização alterada do CNAME descritas nas páginas 313 e 314 não é necessária. Os servidores escravos também estão na vista interna e não lhes é aplicado **chroot**.

```
// arquivo de configuração bind 8.2 – servidor escravo

acl CUnets { 128.138/16; 198.11.16/24; 204.228.69/24; 127.0.0.1; };
acl rfc1918 { 10/8; 172.16/12; 192.168/16; };

options {
   directory "/var/named";
   named-xfer "/usr/local/sbin/named-xfer"; // somente bind 8
   forwarders { 128.138.242.197; };         // endereço IP interno do mestre
   forward first;
   query-source address * port 53;
   allow-transfer { none; };
};

// Registro, dicas de servidor-raiz e zona localhost reversa são os
// mesmos para um servidor para caching, de forma que eles não são mostrados aqui.

// Zonas escravos
zone "cs.colorado.edu" {
   type slave;
   file "forward/cs.colorado.edu";
```

Snow666


```
   masters { 128.138.242.197; };
};

zone "250.138.128.in-addr.arpa" {
     type slave;
     file "reverse/250.138.128";
     masters { 128.138.242.197; };
};

zone "1.168.192.in-addr.arpa" in {          // zonas de endereços privados
      type slave;
      file "reverse/1.168.192";
      allow-query { CUnets; rfc1918; };
      masters { 128.138.242.197; };
};
// … um número muito grande de zonas escravas reversas foi omitido
```

As próximas configurações são para o servidor que é o mestre tanto para as partes interna como externa de cs.colorado.edu, bem como o encaminhador através do qual todas as consultas locais fluem. Essa configuração constrói um belo cache mas quebra a regra "não misture servidores autorizados com servidores para caching".

Essa configuração estabelece uma preferência para os servidores locais através de uma instrução `topology`. Vários servidores não são listados nas delegações do domínio-pai; estes são notificados das modificações com uma cláusula `also-notify`.

O servidor mestre mantém seu banco de dados DNS em vários arquivos. As zonas de mapeamento reverso são organizadas por número de sub-rede. Cada sub-rede (em nosso caso, o terceiro octeto de um endereço classe B) possui seu próprio arquivo. Essa organização não é estritamente necessária, mas ela mantém os arquivos com um tamanho controlável e facilita suas atualizações. Porém, ela pressupõe que as sub-redes sejam divididas num limite de byte ou que, se as sub-redes são subdivididas ainda mais, cada parte permanece sob nosso controle administrativo.

Se um único arquivo fosse usado para todos os mapeamentos reversos, os registros poderiam ser organizados pela rede e a diretiva `$ORIGIN` poderia ser usada no início de cada seção para reiniciar a identidade do domínio-padrão. Veja a seção 16.11.

```
# arquivo de configuração bind 8.x – vista interna, servidor mestre para cs.colorado.edu
# $Id: named.conf,v 1.28 2000/01/12 00:20:34 root Exp $

acl CUnets { 128.138/16; 198.11.16/24; 204.228.69/24; 127.0.0.1; };
acl rfc1918 { 10/8; 172.16/12; 192.168/16; };

# Opções globais
options {
   directory "/";                    # herda o prefixo chroot, /var/named
   pid-file "/named-internal.pid";
   named-xfer "/named-xfer";         # somente V8, programa copiado para /var/named
   notify yes;
   also-notify {
      128.138.192.205;   # suod
      128.138.244.9;     # riker
      128.138.243.70;    # squid
      128.138.241.12;    # goober
      128.138.244.100;   # av-server
      128.138.202.19;    # nago
   };

   # Já que este daemon serve hosts internos, vinculamos o
   # endereço interno, 128.138.242.197 (e localhost).

   listen-on { 127.0.0.1; 128.138.242.197; }; # usa endereços internos
   transfer-source 128.138.242.197;
   query-source address 128.138.242.197 port 53;
   forward first;                    # para acessar zonas externas que
   forwarders { 128.138.243.151; };  # são secundárias para nosso IP externo
   topology { localhost; localnets; CUnets; };
```

Snow666


```
};

# Registro, dicas-raiz e zona localhost são os mesmos e não são mostrados

# CS
zone "cs.colorado.edu" {
   type master;
   file "forward/cs.colorado.edu";
};

# registros reversos CS (128.138.X.X)
zone "250.138.128.in-addr.arpa" {
   type master;
   file "reverse/250.138.128";
};
zone "1.168.192.in-addr.arpa" in {          # redes de espaços de endereços privados
   type master;
   file "reverse/1.168.192";
   allow-query { CUnets; rfc1918; };
};

# … um número muito grande de zonas reversas é omitido

# Escravos
zone "colorado.edu" { # primeiro nível colorado.edu
   type slave;
   file "secondary/colorado.edu";
   allow-transfer { none; };
   masters { 128.138.240.1; };
};
zone "openbsd.org" { # projeto openbsd
   type slave;
   file "secondary/openbsd.org";
   masters { 199.45.131.58; };
};
zone "233.in-addr.arpa" {# endereços multidifusão experimentais
   type slave;
   file "secondary/233.in-addr.arpa";
   masters { 128.223.32.35; };
};

# um número muito grande de zonas é omitido
```

E, finalmente, temos o arquivo de configuração para a vista externa do servidor de nomes mestre para o domínio cs.colorado.edu. Os arquivos de dados internos contêm registros de espaço de endereços privados e também registros MX que apontam para cada host primeiro e para a máquina centralizadora de e-mail como um backup. E-mail interno normalmente flui diretamente e não envolve o centralizador de e-mail. A vista externa dos arquivos de dados inclui um único registro MX para todos os hosts que aponta para a máquina centralizadora de e-mail.

Em vez de repetirmos muitas das mesmas instruções de configuração, listamos a seguir somente aquelas diferentes.

```
# Trechos  da versão externa de config, servidor mestre, cs.colorado.edu

directory "/";                  # servidor externo também recebe o comando chroot para /var/named
pid-file "/named-external.pid";
named-xfer "/named-xfer";
listen-on { 128.138.243.151; }; # endereço IP externo
transfer-source 128.138.243.151;
query-source address 128.138.243.151 port 53;

zone "cs.colorado.edu" in {
   type master;
   file "forward-external/cs.colorado.edu"; # versão externa da zona
```

Snow666


```
};

zone "245.138.128.in-addr.arpa" in {
   type master;
   file "reverse/245.138.128";              # mesmo que a versão interna
};
```

Tanto sobre configuração! Vejamos agora os arquivos reais de dados de zonas.

16.11 O Banco de Dados DNS

Um banco de dados DNS de um domínio é um conjunto de arquivos de texto mantidos pelo administrador de sistemas no servidor de nomes mestre do domínio. Esses arquivos de texto são normalmente chamados de arquivos de zonas. Eles contêm dois tipos de entrada: comandos do analisador sintático (coisas como `$ORIGIN` e `$TTL`) e "registros de recursos" ou RRs como são chamados algumas vezes. Somente os registros de recursos realmente fazem parte do banco de dados; os comandos do analisador sintático simplesmente fornecem algumas maneiras reduzidas para introduzir registros.

Começamos esta seção descrevendo os registros de recursos DNS definidos nas RFCs 882, 1035, 1183, 2065, 2181, 2308 e 2535. Postergamos a discussão dos comandos do analisador sintático para a seção 16.11.

Registros de recursos

Cada zona da hierarquia DNS possui um conjunto de registros de recursos associado a ela (o conjunto pode ser vazio). O formato básico de um registro de recurso é

```
[nome] [ttl] [classe] tipo dados
```

Os campos são separados por espaço em branco (tabulações ou espaços) e podem conter os caracteres especiais mostrados na Tabela 16.9.

O campo *nome* identifica a entidade (normalmente um host ou domínio) que o registro descreve. Se vários registros consecutivos se referirem à mesma entidade, o nome pode ser omitido após o primeiro registro. O campo nome, se presente, deve começar na coluna um.

Um nome pode ser tanto relativo quanto absoluto. Os nomes absolutos terminam com um ponto e são completos. Internamente, o software lida apenas com nomes absolutos; ele anexa o domínio atual e um ponto a qualquer nome que ainda não termine num ponto. Esse recurso permite que os nomes sejam mais curtos, porém ele também é um convite a erros.

Tabela 16.9 Caracteres especiais usados nos RRs.

Caractere	Significado
;	Introduz um comentário
@	O nome de domínio atual
( )	Permite que os dados se estendam sobre linhas
*	Caractere coringa[a] (somente para o campo name)

a. Veja a página 417 para algumas medidas preventivas.

Por exemplo, no domínio cs.colorado.edu domain, o nome "anchor" seria interpretado como "anchor.cs.colorado.edu.". Se o nome fosse introduzido como "anchor.cs.colorado.edu", a ausência de um ponto final ainda implicaria um nome relativo e o domínio-padrão seria anexado, resultando no nome "anchor.cs.colorado.edu.cs.colorado.edu.". Este é um erro cometido comumente.

O campo *ttl* (time to live, tempo de vida) especifica o tempo, em segundos, que o item de dados pode ser colocado em cache e ainda possa ser considerado válido. Normalmente ele é omitido, exceto no arquivo de dicas do servidor-raiz. Seu padrão é o valor configurado pela diretiva `$TTL` logo no início do arquivo de dados para a zona. No BIND 9, a diretiva `$TTL` é exigida. Se não existir nenhuma diretiva `$TTL` no BIND 8, o *ttl* terá como padrão um valor por zona configurado no registro SOA da zona.

Veja o Capítulo 18 para obter maiores informações sobre o NIS.

Aumentar o valor do parâmetro *ttl* para cerca de uma semana reduz substancialmente o tráfego na rede e a carga DNS. Entretanto, assim que os registros tiverem sido colocados em cache fora de sua rede local, você não poderá forçar para que eles sejam descartados. Caso planeje uma renumeração massiva, configure `$TTL` num valor baixo

Snow666

(por exemplo, uma hora) para que os registros velhos que foram colocados em cache em algum ponto na Internet expirem rapidamente.

O DNS funciona bem em larga escala, pois ele usa caching local para distribuir a carga e os dados ao longo da árvore de nomes. Não configure o seu valor de TTL para um valor ridiculamente baixo apenas para manter seus dados atualizados. Com que freqüência você renomeia ou renumera hosts? Provavelmente isso não ocorre a cada segundo, minuto ou até mesmo a cada hora. Mesmo assim, alguns sites — yp.qwestdex.com, por exemplo — configuram o TTL em 5 minutos para que praticamente toda consulta por um número de telefone envolva uma pesquisa DNS completa e, desta forma, acabando com o desempenho de seus sites Web. As unidades politicamente corretas para TTLs são dias ou semanas, e não 300 segundos. Meu Deus!

classe especifica o tipo de rede. São reconhecidos três valores:

- IN de Internet
- CH de ChaosNet
- HS de Hesiod

ChaosNet é um protocolo de rede obsoleto usado anteriormente por máquinas Symbolics Lisp. Hesiod é um serviço de banco de dados construído com base no BIND. O valor-padrão para class é IN. Ele normalmente é especificado explicitamente nos arquivos de dados de zona, muito embora ele seja o padrão. Atualmente, somente um bloco de dados é normalmente guardado na classe Chaos: o número de versão do processo **named** em execução, que pode ser extraído com **dig**, conforme ilustrado na página 276.

São definidos diversos tipos diferentes de registros DNS, mas menos do que 10 são os comumente usados; o IPv6 acrescenta alguns outros. Dividimos os registros de recursos em quatro grupos:

- Registros de zona – identificam domínios e seus servidores de nomes
- Registros básicos – associam nomes a endereços e encaminham e-mail
- Registros de segurança – acrescentam autenticação e assinaturas aos arquivos de zonas
- Registros opcionais – fornecem informações adicionais sobre hosts ou domínios

O conteúdo do campo *dados* depende do tipo de registro. A Tabela 16.10 lista os diversos tipos de registros.

Tabela 16.10 Tipos de registros DNS.

	Tipo	Nome	Função
Zona	SOA	Start Of Authority – Início de Autoridade	Define uma zona DNS
	NS	Name Server – Servidor de Nomes	Identifica servidores de zonas, delega subdomínios
Básicos	A	IPv4 Address	Conversão nome-endereço
	AAAA	Original Ipv6 Address	Era obsoleto, mas está sendo ressuscitado
	A6	IPv6 Address	Conversão nome-endereço-IPv6 (somente para V9)[a]
	PTR	Pointer – Apontador	Conversão endereço-nome
	DNAME	Redirection – Redireção	Redirecionamento para pesquisas IPv6 reversas (somente para V9)
	MX	Mail Exchanger – Servidor de E-mail	Controla o encaminhamento de e-mail
Segurança	KEY	Public Key – Chave Pública	Chave pública para um nome DNS
	NXT	Next – Próximo	Usado com DNSSEC para respostas negativas
	SIG	Signature – Assinatura	Zona autenticada, com assinatura
Opcionais	CNAME	Canonical Name – Nome Canônico	Apelidos ou nomes alternativos para um host
	LOC	Location – Localização	Localização geográfica e extensão[b]
	SRV	Services – Serviços	Dá as localizações de serviços bem conhecidos
	TXT	Text – Texto	Comentários ou informações não transcritas

a. Atualmente considerado "experimental"
b. O regiro LOC não é bem suportado no NT (consultas por registros LOC derrubam servidores NT).

Alguns tipos de registros são obsoletos, experimentais ou não são usados amplamente. Consulte a documentação do BIND para obter uma lista completa.

www.snow666.blogspot.com
Snow666

A ordem dos registros de recursos é praticamente arbitrária. O registro SOA para uma zona deve ser o primeiro. Os registros subseqüentes podem estar em qualquer ordem, porém os registros NS normalmente vêm logo após o SOA. Os registros para cada host normalmente ficam agrupados. É uma prática comum ordenar segundo o campo *nome*.

À medida que formos descrevendo cada tipo de recurso em detalhes, inspecionaremos alguns exemplos de registros dos arquivos de dados de cs.colorado.edu. O domínio-padrão é "cs.colorado.edu." do começo ao fim, de modo que um host especificado como "anchor" na verdade significa "anchor.cs.colorado.edu.".

O registro SOA

Um registro SOA marca o início de uma zona, um grupo de registros de recursos localizados no mesmo local dentro do espaço de nomes DNS. Este nó da árvore DNS também é chamado de ponto de delegação ou corte de zona. Conforme discutiremos com maiores detalhes na página 310, os dados para um domínio DNS normalmente incluem pelo menos duas zonas: uma para converter os nomes de hosts em endereços IP e outras que associam na direção inversa. A árvore DNS possui uma ramificação direta organizada pelo nome e uma ramificação invertida organizada por endereço IP.

Cada zona possui exatamente um registro SOA. A zona prossegue até um outro SOA ser encontrado. O registro SOA inclui o nome da zona, um contato técnico e vários valores de timeout. Um exemplo:

```
; Registro SOA para cs.colorado.edu
@           IN SOA   ns.cs.colorado.edu. hostmaster.cs.colorado.edu. (
                     2001111300   ; Número de série
                     7200         ; Atualização      (2 horas)
                     1800         ; Nova tentativa   (30 minutos)
                     604800       ; Expira           (1 semana)
                     7200 )       ; Mínimo           (2 horas)
```

Nesse caso, o campo *nome* contém o símbolo `@`, que é uma abreviatura para o nome da zona atual. Nesse exemplo, "cs.colorado.edu." poderia ter sido usado em seu lugar. O valor de `@` é o nome de domínio especificado na instrução `zone` no arquivo **named.conf**; ele pode ser alterado de dentro do arquivo de zona por meio de uma diretiva do analisador sintático `$ORIGIN` (veja a página 320).

Esse exemplo não possui nenhum campo *ttl*. A classe é IN de Internet, o tipo é SOA e os itens restantes formam o campo *dados*.

"ns.cs.colorado.edu." é o servidor de nomes mestre da zona.

"hostmaster.cs.colorado.edu." é o endereço de e-mail do contato técnico no formato "*usuario.host.*" e não no formato-padrão *usuario@host*. Simplesmente substitua esse primeiro ponto por um @ e elimine o ponto final caso precise enviar um e-mail para o administrador do domínio. Os sites normalmente usam um nome alternativo como admin ou hostmaster no lugar de um nome de login verdadeiro. O administrador de sistemas responsável pelas tarefas de hostmaster pode mudar e é mais fácil mudar uma entrada no arquivo **aliases** (veja a seção 19.4) do que ter de alterar todos os seus arquivos de zonas quando precisar atualizar a pessoa para contato.

Os parênteses continuam no registro SOA através de várias linhas. Seu posicionamento não é arbitrário no BIND 4 ou 8 — tentamos reduzir a primeira linha dividindo-a antes do endereço para contato, porém isto fez com que **named** não reconhecesse o registro SOA. Em algumas implementações, os parênteses são reconhecidos somente nos registros SOA e TXT. O BIND 9 possui um analisador sintático melhor e os parênteses podem ser usados em qualquer ponto.

O primeiro parâmetro numérico é o número de serie dos dados de configuração da zona. O número de série é usado pelos servidores escravos para determinar quando obter dados atualizados. Ele pode ser qualquer inteiro de 32 bits e deve ser incrementado toda vez que o arquivo de dados para a zona for modificado. Muitos sites codificam a data de modificação do arquivo no número de série. Por exemplo, 2001111300 é a primeira modificação para a zona em 13 de novembro, 2001.

Números de série não precisam ser contínuos, mas devem aumentar monotonicamente. Se, sem querer, você configurar um valor realmente grande no servidor mestre e se esse valor for transferido aos escravos, então corrigir o número de série no mestre não funcionará. Os escravos solicitam dados novos somente se o número de série do mestre for maior do que o deles.

Há três maneiras de corrigir esse problema.

- O BIND 4.9 e o BIND 8 incluem um truque que lhe permite configurar o número de série em zero para um intervalo de atualização e depois reiniciar a numeração. Zero sempre provoca uma recarga; portanto, não se

Snow666

esqueça de configurá-lo para um valor real após cada um dos escravos ter recarregado a zona com o número de série 0.

- Uma forma furtiva porém tediosa de corrigir o problema é modificar o número de série do mestre, eliminar os servidores escravos eliminar os arquivos de dados de backup dos escravos para que eles sejam forçados a recarregar por meio do mestre e reiniciar os escravos.
- Uma terceira maneira de contornar o problema é explorar propriedades do espaço de seqüência no qual os números de série residem. Este procedimento envolve acrescentar um valor grande (2^{31}) para o número de série inchado, permitindo que todos os servidores escravos transfiram os dados e depois configurem o número de série para aquilo que você deseja. Esta estranha aritmética, com exemplos explícitos, é vista com detalhes no livro DNS da O'Reilly; a RFC1982 descreve o espaço de seqüências.

É um erro comum modificar os arquivos de dados, mas esquecer-se de atualizar o número de série. **named** irá puni-lo deixando de propagar suas modificações para os servidores escravos.

As próximas quatro entradas no registro SOA são valores de timeout, em segundos, que controlam por quanto tempo os dados podem ser colocados em cache em diversos pontos ao longo do banco de dados DNS global. Os tempos também podem ser expressos em unidades de minutos, horas, dias ou semanas adicionando-se, respectivamente, um sufixo `m`, `h`, `d` ou `w`. Por exemplo, `1h30m` significa 1 hora e 30 minutos. Os valores de timeout representam um compromisso entre eficiência (é mais barato usar um valor antigo do que buscar um novo) e precisão (valores novos devem ser mais precisos).

O primeiro deles é o timeout *atualizar*, que especifica com que freqüência os servidores escravos devem fazer uma verificação no mestre para ver se o número de série da configuração da zona foi alterado. Toda vez que a zona é alterada, os escravos devem atualizar suas cópias dos dados da zona. O escravo compara os números de série e se o número de série do mestre for maior, solicita uma transferência de zona para atualizar os dados. Valores comuns para o timeout *atualizar* variam de uma a seis horas (3.600 a 21.600 segundos).

Em vez de simplesmente ficar esperando passivamente que os servidores escravos expirem, os servidores BIND agora notificam seus escravos toda vez que uma zona for alterada, a menos que o parâmetro `notify` seja especificamente desativado no arquivo de configuração. Os escravos que entendem a notificação imediatamente se atualizam.

Se um servidor-escravo tentar verificar o número de série do mestre, mas o mestre não responder, o escravo tentará novamente após o período de timeout *nova_tentativa* ter-se expirado. Nossa experiência indica que 20 a 60 minutos (1.200 a 3.600 segundos) é um bom valor.

Se um servidor-mestre estiver inativo por um longo tempo, os escravos tentarão atualizar seus dados muitas vezes, mas sempre falharão. Cada escravo decidirá eventualmente que o mestre jamais responderá e que seus dados certamente estão desatualizados. O parâmetro *expira* determina por quanto tempo os escravos continuarão a servir de forma autorizada os dados do domínio na ausência de um mestre. O sistema deve ser capaz de sobreviver caso o servidor mestre fique fora do ar por uma semana, de modo que esse parâmetro deve ter um valor mais longo. Recomendamos de uma semana a um mês.

O parâmetro *mínimo* no registro SOA configura o tempo de sobrevida de respostas negativas que são colocadas em cache.[12] O padrão para respostas positivas (isto é, registros reais) é especificado logo no início do arquivo de zonas por meio da diretiva `$TTL`. A experiência sugere valores entre algumas horas e vários dias para `$TTL` e de uma a duas horas para o parâmetro *mínimo*. O parâmetro *mínimo* não pode ultrapassar três horas.

Os parâmetros `$TTL`, *expira* e *mínimo* eventualmente forçam todos que usam DNS a descartar valores de dados antigos. O projeto do DNS baseou-se no fato de que dados de host eram relativamente estáveis e não mudavam com freqüência. Entretanto, hosts móveis e DHCP mudaram essas regras. O BIND está tentando desesperadamente acompanhar esse ritmo fornecendo os mecanismos de transferências de zonas incrementais e de atualização dinâmica descritos a partir da página 323. Para maiores informações sobre TTLs e um conceito chamado harmonização TTL, veja a página 320.

Registros NS

Registros NS (name server) identificam os servidores autorizados para uma determinada zona (isto é, todos os servidores mestres e escravos) e delegam subdomínios a outras organizações.

Os registros NS normalmente seguem o registro SOA.

12. Anteriormente ao BIND 8.2, o parâmetro *mínimo* configurava o tempo-padrão de sobrevida para registros de recursos. Ele foi incluído com cada registro e usado para expirar os registros colocados em cache em servidores não autorizados.

Snow666

O formato é

```
zona [ttl] IN NS nome-de-host
```

Por exemplo:

```
cs.colorado.edu.   IN NS ns.cs.colorado.edu.
cs.colorado.edu.   IN NS anchor.cs.colorado.edu.
cs.colorado.edu.   IN NS ns.cs.utah.edu.
```

Já que o nome de zona é o mesmo que o campo *nome* do registro SOA que precede os registros NS, ele pode ser deixado em branco. Portanto, as linhas

```
IN NS ns.cs.colorado.edu.
IN NS anchor.cs.colorado.edu.
IN NS ns.cs.utah.edu.
```

imediatamente após o registro SOA para cs.colorado.edu seriam equivalentes.

Cada servidor de nomes autorizado para cs.colorado.edu que você quer que seja visível externamente deve ser listado tanto no arquivo de zonas para cs.colorado.edu quanto no arquivo para a zona-pai, colorado.edu. Servidores somente para caching não podem ser autorizados; não os liste. Nenhum parâmetro nos registros NS especifica se um servidor é um mestre ou um escravo. Essa informação é especificada no arquivo **named.conf**.

named usa os registros NS de uma zona para identificar servidores escravos quando ele quer enviar notificações para a zona. Esses mesmos registros NS no interior da zona-pai (colorado.edu) definem o subdomínio cs e delegam autoridade sobre ele ao servidor de nomes do departamento de computação. Se a lista de servidores de nomes na zona-pai não for mantida atualizada com aquelas da zona em si, qualquer servidor novo que for adicionado se tornará um servidor invisível e não será usado para responder consultas provenientes do mundo externo. Essa situação acontece algumas vezes devido ao projeto e em outras devido a esquecimento.

Consulte a página 321 para obter maiores informações sobre delegação.

Uma rápida análise em nossas delegações revelaram um servidor principal para colorado.edu sobre o qual o domínio edu não sabia nada. Faça como eu digo, não faça como eu faço: verifique suas delegações com **dig** para ter certeza de que elas especificam um conjunto apropriado de servidores.

Registros A

Os registros A (address) são o cerne do banco de dados DNS. Eles fornecem a associação entre nomes de hosts e endereços IP que foi anteriormente especificada no arquivo **/etc/hosts**. Um host deve ter um registro A para cada uma de suas interfaces de rede. O formato é

```
nome-de-host [ttl] IN A endereço-IP
```

Por exemplo:

```
anchor   IN A     128.138.243.100
```

Uma máquina com várias interfaces de rede pode usar um único nome de host associado a todas as interfaces ou ter nomes de host distintos para cada uma das interfaces.

Registros PTR

Os registros PTR (pointer) realizam a associação reversa de endereços IP para nomes de hosts. Como acontece com os registros A, um host tem de ter um registro PTR para cada interface de rede. Porém, antes de descrevermos os registros PTR, precisamos desviar um pouco do assunto e falar a respeito de um domínio de primeiro nível especial chamado in-addr.arpa.

Nomes de hosts totalmente qualificados podem ser vistos como uma notação na qual a "parte mais significativa" se encontra à direita. Por exemplo, no nome anchor.cs.colorado.edu, anchor se encontra em cs, cs em colorado e colorado em edu. Por outro lado, os endereços IP possuem a "parte mais significativa" à esquerda. No endereço 128.138.243.100, o host 100 se encontra na sub-rede 243, que faz parte da rede 128.138.

O domínio in-addr.arpa foi criado para permitir que um conjunto de módulos de software e uma árvore de atribuição de nomes fizessem a conversão endereços IP-nomes de hosts, bem como de nomes de hosts para endereços IP. Domínios abaixo de in-addr.arpa são nomeados como endereços IP com seus bytes invertidos. Por exemplo, a zona para nossa sub-rede 243 é 243.138.128.in-addr.arpa.

Snow666

O formato genérico de um registro PTR é

```
endereço [ttl] IN PTR nome-de-host
```

Por exemplo, o registro PTR na zona 243.138.128.in-addr.arpa correspondente ao registro A de anchor acima é

```
100          IN PTR anchor.cs.colorado.edu.
```

O nome 100 não termina num ponto e, portanto, é relativo. Mas relativo a quê? Não à "cs.colorado.edu.". Para esse registro de exemplo ser preciso, o domínio-padrão tem de ser "243.138.128.in-addr.arpa.".

Você pode configurar o domínio colocando os registros PTR para cada sub-rede em seu próprio arquivo, como neste exemplo. O domínio-padrão associado ao arquivo é configurado no arquivo de configuração de **named**. Uma outra forma de inverter as associações é incluir registros como

```
100.243      IN PTR anchor.cs.colorado.edu.
```

com um domínio-padrão igual a 138.128.in-addr.arpa. Alguns sites colocam todos os registros invertidos no mesmo arquivo e usam diretivas `$ORIGIN` para especificar a sub-rede. Note que o nome de host anchor.cs.colorado.edu deve terminar com um ponto para impedir que 138.128.in-addr.arpa seja anexado ao seu nome.

Já que cs.colorado.edu e 243.138.128.in-addr.arpa são regiões diferentes do espaço de nomes DNS, elas formam duas zonas distintas. Cada zona deve ter o seu próprio registro SOA e seus próprios registros de recursos. Além de definir uma zona in-addr.arpa para cada rede em si, você também precisa de uma zona que cuide da rede loopback, 127.0.0.0.

Tudo isso funciona bem se as sub-redes estiverem dentro dos limites de byte. Mas como tratar das associações reversas para uma sub-rede como 128.138.243.0/26? Um truque elegante definido na RFC2317 explora os registros de recursos CNAME para realizar esse feito; veja a página 313.

As associações reversas fornecidas por registros PTR são usadas por qualquer programa que autentique tráfego que chega pela rede. Por exemplo, **sshd** pode permitir logins remotos sem uma senha caso a máquina de origem esteja listada, por nome, num arquivo **~/.shosts** de um usuário. Quando o host de destino receber uma solicitação de conexão, ele saberá qual é a máquina de origem somente pelo seu endereço IP. Ele usa DNS para converter o endereço IP num nome de host, que é então comparado ao arquivo apropriado. **netstat**, **tcpd**, **sendmail**, **sshd**, X Windows, **syslogd**, **fingerd**, **ftpd** e **rlogind** fazem associações reversas para obter nomes de hosts com base em endereços IP.

É importante que registros A coincidam com seus registros PTR correspondentes. Registros PTR discordantes e faltantes provocam falhas de autenticação que podem deixar o seu sistema lento como uma lesma. Esse problema é irritante por si só; ele também pode facilitar ataques de negação de serviços contra qualquer aplicativo que requeira a associação reversa para concordar com o registro A.

Veja o Capítulo 19 para obter maiores informações sobre e-mail.

Registros MX

O sistema de e-mail usa registros (MX) de servidores de e-mail para direcionar o e-mail de modo mais eficiente. Um registro MX interpõe-se como destino de uma mensagem, na maioria dos casos direcionando-a para um centralizador de e-mail no site do receptor em vez da estação de trabalho do próprio receptor.

O formato de um registro MX é

```
nome [ttl] IN MX preferência host …
```

Dois exemplos são ilustrados a seguir, um para um host que, a menos que esteja inativo, recebe suas próprias mensagens de e-mail e outro para um host que não é capaz de receber e-mails em si:

```
piper        IN MX 10 piper
             IN MX 20 mailhub
             IN MX 50 boulder.colorado.edu.
xterm1       IN MX 10 mailhub
             IN MX 20 anchor
             IN MX 50 boulder.colorado.edu.
```

Hosts com valores preferenciais baixos são tentados em primeiro lugar: 0 é o mais desejável, e 65.535 é o pior. Nesse exemplo, e-mails endereçados a bob@xterm1 poderiam ser enviados a mailhub caso ele estivesse acessível, para anchor como segunda opção e se tanto mailhub quanto anchor estivessem inativos, para boulder. Note que o nome de boulder tem de ser totalmente qualificado já que ele não é um membro do domínio-padrão (nesse caso, "cs.colorado.edu.").

www.snow666.blogspot.com
Snow666

A lista de preferências e de hosts pode estar tudo na mesma linha, porém linhas separadas facilitam a leitura. Deixe "espaço" numérico entre os valores preferenciais para você não precisar renumerar caso necessite encaixar um novo destino.

Os registros MX são úteis em várias situações:

- Quando se tem um servidor de e-mail centralizado
- Quando o host de destino está inativo
- Quando o destino não pode ser atingido pela Internet
- Quando o host de destino não fala SMTP
- Quando o administrador local sabe para onde o e-mail deve ser enviado melhor do que fazem seus correspondentes

Na primeira dessas situações, o e-mail é direcionado para o centralizador de e-mail, a máquina em que a maioria dos usuários lê seus e-mails. No segundo caso, o e-mail é direcionado para um host próximo e encaminhado quando o destino retorna.

Hosts que não estão diretamente conectados à Internet não podem ter registros A, porém eles podem ter registros MX. **sendmail** não é capaz de se conectar diretamente ao destino, mas ele pode enviar o e-mail para mais próximo conectando-se a um dos hosts MX do destino. Tais hosts somente MX podem ser máquinas por trás de um firewall, nomes de domínios hospedados por um ISP ou serviço de hospedagem, ou então máquinas que não estão ligadas todo o tempo.

A razão final para usarmos registros MX é que os administradores de sistemas locais provavelmente conhecem a arquitetura de e-mail muito melhor do que seus correspondentes. Eles precisam ter a palavra final sobre como o seu site canaliza o seu fluxo de e-mail.

Todo host deve ter registros MX. Para hosts secundários, uma ou duas alternativas são suficientes. Um host principal deve ter vários registros. Por exemplo, o conjunto de registros a seguir poderia ser apropriado para um site no qual cada host envia e recebe seus próprios e-mails.

- Um para o próprio host, como primeira opção
- Um centralizador de e-mail departamental como segunda opção
- Um centralizador de e-mail central para o domínio ou domínio-pai como backup

O próprio domínio deve ter um registro MX para uma máquina centralizadora de e-mail de modo que e-mails para *usuario@dominio* funcionarão. Obviamente, essa configuração realmente requer que os nomes de usuários sejam exclusivos através de todas as máquinas desse domínio. Por exemplo, para sermos capazes de enviar e-mails para evi@cs.colorado.edu, precisamos de uma máquina chamada cs, registros MX em cs.colorado.edu, ou quem sabe de ambos.

```
cs              IN    MX    10 mailhub.cs.colorado.edu.
                IN    MX    20 anchor.cs.colorado.edu.
                IN    MX    50 boulder.colorado.edu.
```

Uma máquina que aceite enviar e-mails para um outro host tem de listar este outro host em seus arquivos de configuração **sendmail**; veja a seção 19.8 para uma discussão sobre recurso `use_cw_file` do **sendmail** e do arquivo **local-host-names**.

Registros MX couringa também são algumas vezes vistos no banco de dados DNS:

```
*               IN MX 10 mailhub.cs.colorado.edu.
```

À primeira vista, esse registro parece capaz de poupar um bocado de digitação e de adicionar um registro MX padrão para todos os hosts. Porém, os registro curinga não funcionam bem como esperamos. Eles concordam com qualquer coisa no campo *nome* de um registro de recursos que já *não* esteja listado como um nome explícito num outro registro de recursos.

Portanto, você *não pode* usar um asterisco para configurar um valor-padrão para todos os seus hosts. Mas, de modo contrário, você pode usá-lo para configurar um valor-padrão para nomes que não são seus hosts. Essa configuração faz com que muitas mensagens de e-mail sejam enviadas para o seu centralizador somente para ser rejeitadas, pois o nome de host que concorda com o asterisco realmente não pertence ao seu domínio. Portanto, evite registros MX curinga. Os couringas ainda não são implementados no BIND 9 e talvez jamais o sejam.

Snow666

Registros CNAME

Os registros CNAME atribuem nomes adicionais a um host. Esses apelidos são comumente usados tanto para associar uma função a um host como para reduzir um nome de host longo. O nome real algumas vezes é chamado de nome canônico (daí, "CNAME").

Alguns exemplos:

```
ftp         IN CNAME anchor
kb          IN CNAME kibblesnbits
```

O formato de um registro CNAME é

```
apelido [ttl] IN CNAME nome-de-host
```

Quando o software DNS encontra um registro CNAME, ele interrompe sua consulta em busca do apelido e muda para o nome real. Se um host possui um registro CNAME, outros registros (A, MX, NS etc.) para este host têm de se referir ao seu nome real, e não ao seu apelido. Por exemplo, as linhas a seguir são corretas.

```
colo-gw     IN A     128.138.243.25
moogie      IN CNAME colo-gw
www         IN CNAME moogie
```

Entretanto, atribuir uma prioridade de endereço ou de e-mail (com um registro A ou MX) seja para www como para moogie nesse exemplo seria incorreto.

Os registros CNAME podem ser aninhados em até oito níveis no BIND. Isto é, um registro CNAME pode apontar para um outro CNAME e este CNAME pode apontar para um terceiro CNAME e assim por diante, até no máximo sete vezes; o oitavo destino deve ser o nome de host real e o registro A.

Alguns sites usam registros CNAME numa frágil tentativa de balanceamento de carga. Eles associam o nome público de seu servidor Web a várias máquinas diferentes:

```
www         IN CNAME web1
www         IN CNAME web2
www         IN CNAME web3
```

Esse emprego de registros CNAME não é padronizado. De fato, ele é ilegal. Uma opção no BIND 8 permitia que usássemos esse mecanismo contrário às especificações. O BIND 9 é mais chato, de modo que realmente não é uma boa idéia usar vários CNAMEs. Uma melhor maneira de conseguir o mesmo resultado é prover o servidor Web com vários registros A que apontam para máquinas diferentes.

Consulte a seção 13.4 para obter maiores informações sobre CIDR.

O truque CNAME

Os CNAMEs também são usados para forçar a semântica existente do DNS a suportar zonas reversas para redes que não estão em sub-redes num limite de byte. Antes de o endereçamento CIDR ser um lugar comum, a maioria das atribuições de sub-rede era em limites de byte ou dentro da mesma organização e as delegações reversas eram fáceis de serem gerenciadas. Por exemplo, se a rede classe B 128.138 fosse subdividida em sub-redes tipo classe C, cada sub-rede seria um bloco organizado para o domínio in-addr.arpa. A zona reversa para a sub-rede 243 seria 243.138.128.in-addr.arpa.

Mas o que acontece se a sub-rede 243 for ainda mais dividida, digamos, em quatro trechos como uma rede /26? Se todos os quatro trechos forem atribuídos à mesma organização, na verdade, não há nenhum problema. As quatro sub-redes ainda poderão compartilhar um único arquivo contendo todos os seus registros PTR. Porém, se a sub-rede 243 for atribuída a um ISP que quer delegar cada rede /26 a um cliente diferente, se faz necessária uma solução mais complexa. O ISP tem de manter os registros reversos em nome de cada cliente ou então tem de encontrar uma maneira de pegar o terceiro octeto do endereço IP (243 nesse caso) e dividi-lo em quatro trechos diferentes que podem ser delegados independentemente.

Quando um limite administrativo localiza-se no meio de um byte, você terá de ser habilidoso. Você também terá de trabalhar em sintonia com o domínio acima ou abaixo de você. O truque é o seguinte: para cada endereço de host possível na zona in-addr.arpa natural, acrescente um CNAME que desvie a procura para uma zona controlada pelo proprietário da sub-rede adequada. Esse esquema produz arquivos de zonas confusos no pai, porém ele permite que você delegue autoridade aos verdadeiros usuários de cada sub-rede.

Eis o esquema em detalhes. A organização-pai (em nosso caso, o ISP) cria registros CNAME para cada endereço IP possível com um componente dissimulador adicional (trecho separado por ponto) que representa a sub-rede.

Snow666

Por exemplo, na situação da rede /26 descrita há pouco, o primeiro quarto dos endereços teria um componente "0-63", o segundo quarto teria um componente "64-127" e assim por diante. Eis como ele ficaria:

```
$ORIGIN 243.138.128.in-addr.arpa.
1           IN CNAME 1.0-63
2           IN CNAME 2.0-63
...
63          IN CNAME 63.0-63
64          IN CNAME 64.64-127
65          IN CNAME 65.64-127
...
```

Para delegarmos o trecho 0-63 da zona reversa ao cliente que foi designado àquela sub-rede, acrescentamos os seguintes registros NS:

```
0-63        IN NS    ns1.customer1.com.
0-63        IN NS    ns2.customer1.com.
...
```

O site de customer1.com teria um arquivo de zonas que conteria as associações reversas para a zona 0-63.243.138.128.in-addr.arpa. Por exemplo:

```
1           IN PTR   host1.customer1.com.
2           IN PTR   host2.customer1.com.
...
```

Acrescentando um componente extra, criamos um novo "corte" no qual podemos realizar a delegação. Quando, por exemplo, alguém procurar a associação reversa para 128.138.243.1, o registro CNAME em 1.243.138.128.in-addr.arpa redirecionará a pesquisa para o nome 1.0-63.243.138.128.in-addr.arpa, e este nome será controlado pelo cliente.

Os arquivos do cliente são organizados; somente o ISP que deve lidar com aquela confusão de configuração. Mas as coisas podem se tornar ainda mais complicadas. O próprio Customer1 poderia ser um ISP que quisesse subdividir ainda mais seus endereços. Tudo bem: o BIND suporta cadeias de CNAME de até 8 links de profundidade e, já que um byte possui apenas oito bits, jamais poderemos ultrapassar este limite. O emprego de cadeias de CNAMEs não é encorajado, porém isto não é proibido pelas RFCs; elas realmente desaceleram o processo de determinação de nomes, pois cada link numa cadeia CNAME faz com que o link seja seguido e uma nova consulta para o destino seja iniciada.

Todo esse esquema é um abuso espalhafatoso do registro CNAME, mas ele é tão poderoso e útil que uma variante dele se tornou o padrão oficial para manipulação de zonas reversas IPv6. Veja as informações sobre registros DNAME na seção 16.11 para obter maiores informações.

Logo que o truque do CNAME começou a ser utilizado, o comando `$GENERATE` foi acrescentado ao repertório de **named** para facilitar a criação de registros de recursos na zona-pai.

Por exemplo, para gerar os registros para a primeira sub-rede, as linhas a seguir já seriam suficientes:

```
$ORIGIN 243.138.128.in-addr.arpa.
$GENERATE 0-63 $ CNAME $.0-63
0-63        NS ns1.customer1.com.
0-63        NS ns2.customer1.com.
```

O `$` no comando `$GENERATE` varia de 0 a 63 e cria 64 registros CNAMEs diferentes. As outras três redes /26 seriam manipuladas de maneira similar.

O truque CNAME funciona bem para o BIND 8 e 9. Alguns solucionadores BIND 4 não esperam obter um CNAME quando eles consultam um registro PTR, de forma que eles falham. Mais uma boa razão para fazer a atualização.

Os registros LOC são definidos na RFC1819.

Registros LOC

Um registro LOC descreve a localização geográfica e, opcionalmente, o tamanho físico (diâmetro) de um objeto DNS. Os registros LOC atualmente não possuem nenhum efeito sobre a operação técnica da Internet e nenhum software-padrão os pesquisa. Porém, uma série de empregos potencial interessantes para as informações foi sugerida, inclusive rastreio e otimização de rotas, mapeamento automatizado e pesquisa de redes.

Snow666

O formato é:

nome [*ttl*] IN LOC *lat lon* [*alt* [*tamanho* [*hp* [*vp*]]]]

A latitude e longitude são dadas na forma de graus, minutos e segundos separados por espaços seguidos por N, S, E ou W. Os segundos podem ser omitidos; se forem, os minutos também podem ser omitidos.

Os demais campos são todos especificados em centímetros (nenhum sufixo) ou metros (m). *alt* é a altitude do objeto, *tamanho* é o diâmetro da esfera que circunda o objeto, *hp* é a precisão horizontal da medida e *vp* a precisão vertical. O tamanho-padrão é de um metro, e as precisões horizontal e vertical padrão são de, respectivamente, 10 metros e 10 quilômetros.

Eis um exemplo para caida.org em San Diego, Califórnia:

```
caida.org. IN LOC 32 53 01 N 117 14 25 W 107m 30m 18m 15m
```

Muitas das ferramentas para visualização gráfica escritas pelo CAIDA (Cooperative Association for Internet Data Analysis) requerem dados de latitude e de longitude e os sites são encorajados a incluir estas informações em seus DNSs. Entretanto, se você for paranóico e rodar um servidor de alta visibilidade ou ISP, talvez não queira que o público em geral saiba a localização exata de suas máquinas. Em tais situações, recomendamos que você use valores inexatos com um grande parâmetro de precisão horizontal. Registros LOC imprecisos ainda são de valia para o pessoal de pesquisa em redes, mas oferecem algum anonimato.

Um outro recurso notável dos registros LOC é que eles parecem derrubar servidores de nomes NT 4.0; tome precauções.

Registros SRV

Um registro SRV especifica a localização de serviços dentro de um domínio. Por exemplo, o registro SRV permite que você consulte um domínio remoto diretamente e pergunte pelo nome de seu servidor FTP. Até agora, grande parte é adivinhação. Para contactar o servidor FTP para um domínio remoto, você tinha de torcer que os administradores remotos seguissem o costume atual e adicionassem um CNAME para "ftp" para registros DNS de seus servidores.

Os registros SRV fazem mais sentido do que os CNAMEs para essa aplicação e são, certamente, uma melhor maneira para administradores de sistemas deslocarem serviços e controlarem seus empregos. Porém, os registros SRV têm de ser pesquisados explicitamente e analisados sintaticamente pelos clientes, de modo que levará um tempo antes de seus efeitos realmente serem sentidos.

Os registros SRV lembram registros MX genéricos com campos que permitem ao administrador DNS local dirigir e balancear a carga de conexões do mundo externo. O formato é

serviço.protocolo.nome [*ttl*] IN SRV *pri wt porta alvo*

onde *serviço* é um serviço definido no banco de dados de números atribuídos IANA (veja a página 185 ou www.iana.org/numbers.htm), *protocolo* é tcp ou então udp, *nome* é o domínio para o qual o registro SRV se refere, *pri* é uma prioridade estilo MX, *wt* é um peso usado para balanceamento de cargas entre diversos servidores, *porta* é a porta na qual o serviço é executado e *alvo* é o nome de host do servidor que fornece este serviço. O registro A do destino normalmente é retornado de maneira automática com a resposta a uma consulta SRV. Um valor 0 para o parâmetro *wt* significa que não precisa ser feito nenhum balanceamento de carga especial. Um valor "." para o alvo significa que o serviço não é executado nesse site.

Eis um exemplo, extraído da RFC2052 (onde SRV é definido) e adaptado para o domínio cs.colorado.edu:

```
ftp.tcp         SRV 0 0 21 ftp-server.cs.colorado.edu.

; não permite mais conexões finger (destino = .)
finger.tcp      SRV 0 0 79 .

; 1/4 das conexões para o computador antigo, 3/4 para o novo
ssh.tcp         SRV 0 1 22 old-slow-box.cs.colorado.edu.
                SRV 0 3 22 new-fast-box.cs.colorado.edu.

; servidor principal na porta 80, backup no novo computador, porta 8000
http.tcp        SRV 0 0 80 www-server.cs.colorado.edu.
                SRV 10 0 8000 new-fast-box.cs.colorado.edu.

; portanto, tanto http://www.cs.colo… quanto http://cs.colo… funcionam
http.tcp.www    SRV 0 0 80 www-server.cs.colorado.edu.
```

Snow666


```
                SRV 10 0 8000 new-fast-box.cs.colorado.edu.

; bloqueia todos os demais serviços (destino = .)
*.tcp           SRV 0 0 0 .
*.udp           SRV 0 0 0 .
```

Esse exemplo ilustra o emprego tanto do parâmetro de peso (para SSH) quanto do parâmetro de prioridade (HTTP). Ambos os servidores SSH serão usados, com o trabalho dividido entre eles. O servidor HTTP de backup será utilizado somente quando o servidor principal não estiver disponível. O serviço **finger** não é incluso, assim como não são outros serviços não mencionados explicitamente. O fato de que o daemon **finger** não aparece no DNS não significa que ele não esteja em execução, mas sim que você não foi capaz de localizar o servidor através do DNS.

WKS (Well-Known Services) era um antigo registro DNS relacionado com serviços que não pegou. Em vez de apontar você para o host que fornecia um determinado serviço para um domínio, ele listava os serviços fornecidos por um determinado host. O WKS parecia ser um tanto inútil e também era considerado um risco em termos de segurança. Ele não foi amplamente adotado.

Registros TXT

Um registro TXT acrescenta texto arbitrário aos registros DNS de um host. Temos, por exemplo, um registro TXT que identifica nosso site:

```
IN              TXT "University of CO, Boulder Campus, CS Dept"
```

Esse registro vem logo após os registros SOA e NS para a zona "cs.colorado.edu." e também herda o campo *nome* deles.

O formato de um registro TXT é o seguinte:

```
nome [ttl] IN TXT informação …
```

Todos os itens *informação* têm de ser colocados entre aspas. Você pode usar uma string envolta por um par de aspas ou então várias strings, em que em cada uma são colocadas aspas. Certifique-se de que as aspas estejam balanceadas — a falta de uma simples aspas corromperá os seus dados DNS, pois todos os registros entre as aspas faltantes e a próxima ocorrência de aspas desaparecerão misteriosamente.

Os registros TXT não possuem nenhuma ordem intrínseca. Se você usar vários deles para acrescentar um parágrafo de informações ao seu DNS, eles podem acabar ficando misturados quando **named** e UDP já os tiverem utilizado.

Veja o Capítulo 13 para obter uma discussão detalhada sobre o IPv6.

Registros de recursos no IPv6

IPv6 é uma nova versão do protocolo IP. Está há aproximadamente 10 anos no processo de especificação e ainda não está completo. O IPv6 foi originalmente motivado por uma necessidade de mais números de endereços de rede IP. Entretanto, as soluções improvisadas para esse problema — CIDR, NAT e controle de endereços mais rígido — foram tão bem-sucedidas que é pouco provável que ocorra uma migração em massa para o IPv6 tão cedo. A menos que alguém surja com novo aplicativo arrasador que rode somente no IPv6 (ou que alguma versão futura do Microsoft Windows o tenha como padrão), dificilmente os administradores de sistemas terão de lidar com IPv6 por alguns anos mais. Algumas pessoas acham que a próxima geração de telefones celulares, que poderão vir a ter endereços IP, pode simplesmente pesar na balança a favor do IPv6.

Muito embora não esperemos ver o IPv6 empregado tão cedo, acreditamos que vale a pena descrever o impacto de endereços IP de 128 bits sobre o sistema DNS. Tanto os registros de endereços quanto os registros de ponteiro têm de mudar, porém estas alterações são relativamente simples comparadas à tarefa de suportar um dos conceitos totalmente novos do Ipv6: propriedade compartilhada de endereços.

A interface de host a qual um endereço IPv6 corresponde possui alguns dos bits de endereços, mas não todos eles. Outros bits são delegados aos ISPs acima do site numa tentativa de facilitar a tarefa de renumeração e de alteração de ISPs. Esse projeto acrescenta um bocado de complexidade. Após observarmos todas as dificuldades que o DNS teve de passar para suportar os padrões, imaginamos se os autores destes padrões criaram algum programa recentemente. Provavelmente não.

Os equivalentes no IPv6 dos registros A do DNS foram originalmente chamados de registros AAAA, pois os endereços IPv6 eram quatro vezes mais longos do que os endereços IPv4. Como o esquema de controle de subdivisão para endereços IPv6 evoluiu, o IETF tornou os registros AAAA obsoletos e padronizou dois novos tipos de registros: registros A6 para associações nome-endereço e registros DNAME para delegação das porções de endereço para diferentes organizações. Os registros DNAME se inspiraram no truque CNAME (veja a seção16.11), que permite a

Snow666

delegação em limites de bits. Os registros A6 especificam um endereço, mas com a possibilidade de que alguns bits de prefixo de ordem mais alta tenham de ser obtidos de uma outra fonte.

Na metade de 2001, o IETF, frustrado com o baixo emprego do IPv6, se deu conta de que a complexidade dos endereços compartilhados com registros DNAME e A6 era o que estava emperrando o processo. Eles transferiram A6 e DNAME do estado-padrão para o estado experimental e readmitiram os registros de endereço AAAA e a forma esmiuçada do registro PTR. Ainda há controvérsia em relação ao melhor formato de endereço. AAAA é (muito) mais simples, mas destrói a renumeração fácil, que foi anteriormente um objetivo-chave do IPv6.

Já que não esperamos que o IPv6 seja empregado largamente tão cedo, seja na forma AAAA como na forma A6, adiamos as descrições detalhadas dos mecanismos de pesquisa do IPv6 até serem melhor definidas pelo IETF e tivermos alguma experiência operacional com elas. Por enquanto, as seções a seguir dão uma visão geral dos dois planos. Você pode ir direto para a seção *Comandos em arquivos de zonas*, caso prefira não ler a respeito do IPv6.

Registros de endereços IPv6

O formato de um registro AAAA é

```
nome-de-host [ttl] IN AAAA endereço-IP
```

Por exemplo:

```
anchor      IN AAAA  3ffe:8050:201:9:a00:20ff:fe81:2b32
```

Cada trecho separado por dois-pontos corresponde a 4 dígitos hexadecimais, com zeros não significativos normalmente omitidos. Dois-pontos adjacentes significam "zeros suficientes para preencher os 128 bits para um endereço IPVv6 completo". Um endereço pode conter no máximo um destes dois-pontos duplos.

O formato de um registro A6 é similar, mas com uma pequena alteração:

```
nome-de-host [ttl] IN A6 num-bits-delegados endereço-IP delegado
```

Por exemplo:

```
anchor      IN A6    0 3ffe:8050:201:9:a00:20ff:fe81:2b32 .
anchor      IN A6    48 ::9:a00:20ff:fe81:2b32 prefix.myisp.net.
```

Esses dois registros A6 especificam o mesmo endereço IPv6 para o host anchor; um é completamente especificado sem nenhum bit de prefixo delegado e o outro possui 48 bits de prefixo delegados ao host prefix.myisp.net. Veja o ponto como o parâmetro delegado na primeira forma do registro A6; ele significa que nenhum delegado adicional é necessário.

Pesquisas de nomes A6 diretos podem ter de falar com vários servidores de nomes subindo a cadeia A6 para montar um endereço de 128 bits completo. Por exemplo, com a segunda linha acima, o próximo nível poderia delegar 47 bits, o seguinte poderia delegar 46 bits e assim por diante. Talvez sejam necessárias 48 consultas para obter a resposta completa. Adicione a esse número as consultas DNSSEC necessárias para verificar cada trecho do endereço, e você terá um aumento de 100 vezes no tráfego DNS para descobrir um nome. Os projetos realizados pelos comitês nem sempre são simples e eficientes. Esses 48 níveis em potencial irão, na prática, mais provavelmente ser 2 ou 3, porém o conceito pode apresentar interessantes avenidas para ataques de negação de serviços. Para uma documentação mais completa (e menos preconceituosa), consulte o diretório **doc** do BIND 9.

Registros reversos no IPv6

No IPv4, as associações reversas residem no domínio in-addr.arpa e as associações diretas residem nas demais ramificações da árvore de domínios (sob com ou edu, por exemplo). No Ipv6, as informações de associação reversas correspondentes a um registro de endereço AAAA são um registro PTR no domínio de primeiro nível ip6.int. O registro reverso para um registro de endereço A6 é um pouco mais disperso. Alguns deles vivem sob o domínio ip6.arpa e o restante é armazenado entre os domínios diretos.

O formato esmiuçado inverte um registro de endereço AAAA expandindo cada trecho de endereço separado por dois-pontos para os 4 dígitos hexadecimais completos e depois invertendo a ordem destes dígitos e anexando ip6.int ao final. Por exemplo, o registro PTR correspondente ao nosso exemplo de registro AAAA para anchor ficaria

```
2.3.b.2.1.8.e.f.f.f.0.2.0.0.a.0.9.0.0.0.1.0.2.0.0.5.0.8.e.f.f.3.ip6.int PTR anchor.cs.colorado.edu.
```

Certamente não é agradável para um administrador de sistemas ter de digitar ou depurar ou até mesmo ler. Obviamente, em seus arquivos de zonas DNS reais, a instrução `$ORIGIN` ocultaria parte da complexidade.

Entretanto, o formato concorrente para pesquisas reversas faz com que esse formato esmiuçado se pareça, comparativamente, com uma tarefa de matemática de aluno da 1ª série. O formato alternativo reside numa ramifica-

www.snow666.blogspot.com
Snow666

ção de primeiro nível da árvore de nomes DNS, ip6.arpa, e usa tanto registros PTR tradicionais quanto registros de um novo tipo específico ao IPv6, DNAME. Os registros PTR associam os bits locais de um endereço IPv6 a um determinado nome de host e os registros DNAME determinam que partes do restante do endereço são delegadas a quais organizações.

Os componentes dos nomes na hierarquia in-addr.arpa representam os bytes de um endereço IP. Para IPv6, generalizamos esse esquema e permitimos que componentes de nomes representem seções arbitrárias de um endereço. As seções de endereço podem ter qualquer número de bits de largura (para valores de "qualquer número" entre 1 e 128) e são conhecidas como bitstrings (strings de bits ou seqüências de bits).

Bitstrings são representadas com uma sintaxe peculiar conhecida como um rótulo bitstring. Vejamos um exemplo. Todos os endereços A6 unidifusão IPv6 começam com os três bits 001. Para expressarmos esse prefixo na linguagem de bitstrings, começamos com o número binário 001 e o preenchemos para formar um múltiplo de quatro bits: 0010. Esse cálculo nos dá o dígito hexadecimal 2; o dígito possui três bits válidos e um bit de descarte. O bitstring final é:

```
\[x2/3]
```

A barra invertida, colchetes e x delimitam cada bitstring. As partes importantes são uma série de dígitos hexadecimais (somente um nesse caso, 2) e o qualificador de comprimento, /3. O qualificador de comprimento, que informa quantos dos bits representados pelos dígitos hexadecimais são realmente válidos, é opcional. Se omitido, a bitstring tem como padrão seu comprimento natural conforme determinado pelo número de dígitos hexadecimais (nesse caso, 4 bits).

Mesmo se suas bitstrings terminarem num limite de um dígito hexadecimal, é uma boa idéia incluir o qualificador de comprimento. Caso contrário, os leitores de seus arquivos DNS ficarão cegos ao contarem strings longas de minúsculos dígitos hexadecimais.

Os bits mais à esquerda de uma string hexadecimal são os significativos. Bits extras usados para preencher o dígito hexadecimal mais à direita são simplesmente descartados. Os bits de preenchimento têm de ser 0s.

Pelo fato de os endereços IPv6 unidifusão compartilharem o mesmo prefixo 001, o domínio de primeiro nível efetivo para as associações reversas A6 é \[x2/3].ip6.arpa.

Eis um exemplo mais completo. As linhas a seguir ilustram três representações diferentes do mesmo endereço: o primeiro não dividido, o segundo dividido em três trechos (3/45/80 bits) e o terceiro dividido em quatro trechos (3/13/32/80).[13] Como trechos de endereços são deslocados, suas representações hexadecimais mudam completamente — entretanto, são os mesmos bits que estão lá internamente. **bc** é o seu amigo para deslocamento de bits.[14]

```
\[x3ffe8050020100090a0020fffe812b32/128].ip6.arpa.
\[x00090a0020fffe812b32/80].\[xfff402801008/45].\[x2/3].ip6.arpa.
\[x00090a0020fffe812b32/80].\[x80500201/32].\[xfff0/13].\[x2/3].ip6.arpa
```

Como acontece com zonas in-addr.arpa IPv4, números individuais são lidos da esquerda para a direita e componentes (trechos separados por pontos) são lidos da direita para a esquerda. O primeiro componente na segunda e terceira linhas acima é a parte local do endereço. Ele representa os 80 bits de ordem inferior do endereço e é formado pelos dígitos hexadecimais 00090a0020fffe812b32. Abaixo temos as mesmas três linhas com a parte local do endereço em negrito:

```
\[x3ffe8050020100090a0020fffe812b32/128].ip6.arpa.
\[x00090a0020fffe812b32/80].\[xfff402801008/45].\[x2/3].ip6.arpa.
\[x00090a0020fffe812b32/80].\[x80500201/32].\[xfff0/13].\[x2/3].ip6.arpa
```

O /3 na segunda linha indica que os três primeiros dos quatro bits do dígito hexadecimal 2 são partes válidas do endereço. O /45 na segunda linha significa que os primeiros 45 bits dos 48 bits presentes na string hexadecimal fff402801008 são válidos no trecho intermediário do endereço. Todos esses fragmentos vão ser fáceis de serem digitados em seus arquivos DNS sem erros, não é mesmo? Minha nossa!

Os registros DNAME delegam trechos de endereços a outros servidores de nomes. O formato deles é o seguinte:

```
rótulo-da-bitstring [ttl] IN DNAME domínio-delegado
```

13. Enquanto se discute em nível de bits, é fácil esquecer o fato de que endereços IPv6 possuem alguma estrutura interna própria. Consulte a seção 13.4 para uma discussão dos limites e significados das regiões nas quais um endereço IPv6 é dividido convencionalmente.
14. **bc**, o utilitário para cálculos do Linux, inclui o conceito de bases numéricas; você pode especificar a base numérica de entrada com ibase e a base numérica de saída com obase. Por exemplo, ibase=16 e ibase=2 transformariam qualquer número digitado em **bc** de hexadecimal para binário. Assim que você tiver mudado de base de entrada, será mais fácil reiniciar o **bc** para retornar ao normal, pois **bc** interpreta todas as entradas de acordo com a base atual, mesmo comandos ibase novos.

Snow666

A idéia é a de que organizações diferentes possam controlar os diferentes trechos de um endereço ou prefixo de endereço. Você controla os 80 bits locais, seu ISP controla um trecho, o ISP de seu ISP controla um outro trecho e assim por diante. Os trechos a seguir demonstram a delegação que os registros DNAME são capazes de realizar. Usamos o endereço acima em sua versão de quatro trechos, no qual o caminho de delegação é da raiz para my-isp para my-domain.

Neste exemplo, as instruções `$ORIGIN` identificam o contexto de cada registro. Esses trechos são de arquivos de zonas em três sites diferentes: a raiz da árvore ip6.arpa, my-isp.net e my-domain.com — eles não representam uma configuração consistente para um único site.

A raiz da árvore ip6.arpa, \[x2/3].ip6.arpa, delega um determinado endereço de 13 bits a my-isp.net introduzindo as seguintes linhas em seu arquivo de zonas:

```
; delega prefixo a my-isp.net
$ORIGIN \[x2/3].ip6.arpa.
\[xfff0/13]        IN DNAME ip6.my-isp.net.
```

Esses registros criam uma espécie de apelido para \[xfff0/13].\[x2/3].ip6.arpa que aponta para a string "ip6.my-isp.net.". O ISP, por seu lado, delega um segmento de 32 bits de espaço de endereços a my-domain.com incluindo as linhas a seguir ao seu arquivo de zonas ip6.my-isp.net:

```
; delega prefix a my-domain.com
$ORIGIN ip6.my-isp.net.
\[x80500201/32] IN DNAME ip6.my-domain.com.
```

Essa linha forma um nome, "\[x80500201/32].ip6.my-isp.net.", que quando expandido a partir do registro DNAME anterior se torna o prefixo de 48 bits do endereço IPv6. Esses 48 bits ganham o apelido ip6.my-domain.com.

Nos arquivos de zonas para ip6.my-domain.com, o restante do endereço é associado a um registro PTR:

```
$ORIGIN ip6.my-domain.com.
\[x00090a0020fffe812b32/80]        IN PTR   host.my-domain.com.
```

Sugerimos que você não se preocupe com as delegações DNAME caso tenha somente um ISP e não tenha mais de uma interface de rede. Liste simplesmente o endereço completo de 128 bits tanto nas zonas direta quanto reversa. Você ainda obtém alguns bits de seu ISP acima, mas pode colocá-los em seus próprios arquivos de zonas desde que seu ISP informe-o quando mudar o prefixo.

O IPv6 ainda é jovem, pelo menos sob a perspectiva de emprego. Os escritórios de registros estão apenas começando a atribuir endereços e o processo se tornará mais fluente à medida que se for acumulando experiência. A parte local do endereço IPv6 jamais precisa ser modificada e isto é ótimo — nenhuma renumeração. Mas em várias organizações, ISPs acima também não mudam com tanta freqüência e, portanto, colocar os bits do ISP em seus próprios arquivos de dados parece ser uma boa coisa. Um pequeno script em Perl poderia alterar o prefixo de todos os seus endereços caso alguma vez você venha a trocar de ISP. Assim que o IETF se convencer sobre qual esquema ele realmente usará, podemos considerar a utilização. Porém, certamente existirão mais alterações, de modo que é realmente muito cedo para começar a usar o IPv6 no DNS. Os servidores-raiz ainda não falavam IPv6 no início de 2002, nem manipulavam delegações nos domínios ip6.int ou ip6.arpa. Os servidores gTLD estão pensando em adotar os registros AAAA do IPv6 em breve[15].

Comandos em arquivos de zonas

Agora que vimos todos os registros de recursos básicos, vamos dar uma olhada nos comandos que podem ser embutidos num arquivo de zonas para modificar os registros que os seguem. Há quatro deles:

```
$ORIGIN nome-do-domínio
$INCLUDE nome-de-arquivo
$TTL ttl-padrão
$GENERATE vários-argumentos
```

15. A adoção de endereços IPv6 está se consolidando, mas o seu emprego em larga escala na Internet ainda não aconteceu. Algumas questões de padronização ainda não foram resolvidas com relação aos registros DNS; vide RFC 3363. Entretanto, há um forte movimento de fabricantes de hardware, provedores de serviços de internet (ISP), governos e outros, que tenta provocar a adoção do IPv6. (N.R.T.)

Snow666

Os comandos têm *obrigatoriamente* que começar na primeira coluna e cada um deles deve ocupar uma linha exclusiva.

À medida que **named** lê um arquivo de zonas, ele adiciona o domínio-padrão (ou "origem") a qualquer nome que não esteja ainda completamente qualificado. A origem é configurada inicialmente ao nome de domínio especificado na instrução `zone` correspondente em **named.conf**. Porém, você pode configurar a origem manualmente dentro de um arquivo de zonas usando a diretiva `$ORIGIN`.

O emprego de nomes relativos em que se espera nomes totalmente qualificados poupa um bocado de digitação e torna os arquivos de zonas muito mais fáceis de serem lidos. Por exemplo, os registros reversos para um site classe B subdividido em redes poderiam estar todos num único arquivo de zonas, com as instruções `$ORIGIN` configurando o contexto para cada sub-rede. Uma instrução como

```
$ORIGIN 243.138.128.in-addr.arpa
```

poderia anteceder os registros para s sub-rede 243.

Muitos sites usam a diretiva `$INCLUDE` em seus arquivos de bancos de dados de zonas para separar registros extras de registros de dados, para separar trechos lógicos de um arquivo de zonas ou para manter chaves criptográficas num arquivo com permissões restritas. A sintaxe da diretiva `$INCLUDE` é

```
$INCLUDE nome-de-arquivo
```

O arquivo especificado é lido no banco de dados no ponto da diretiva `$INCLUDE`. Se *nome-de-arquivo* não for um caminho absoluto, ele será interpretado relativamente ao diretório especificado no arquivo **named.conf** como o diretório-base de **named**.

A diretiva `$TTL` configura um valor-padrão para o tempo de sobrevida dos registros que o seguem. Antigamente, a única maneira de configurar o padrão era no registro SOA (descrito na página 307). O BIND 8 gosta de ter uma diretiva `$TTL` no início de arquivos de zonas; o BIND 9 o exige e se recusa a carregar arquivos de zonas que não configurem um `$TTL` padrão.

A instrução `$TTL` deve anteceder o registro SOA para a zona. As unidades-padrão para o valor `$TTL` são segundos, mas você também pode qualificar números com h para horas, m para minutes, d para dias ou w para semanas. Por exemplo, todas as linhas a seguir

```
$TTL 86400
$TTL 24h
$TTL 1d
```

configuram `$TTL` para um dia.

O valor de `$TTL` deve ser da ordem de dias ou semanas para permitir que o caching ajude o DNS a aumentar para uma Internet cada vez mais em expansão. Valores válidos vão de 0 a 2.147.483.647 segundos (cerca de 65 anos). Isso é muito tempo! Se você estiver planejando mudar máquinas e souber que terá de modificar nomes de hosts ou endereços IP de sua "frota" existente, configure TTL para um valor mais baixo (digamos, uma hora), uma semana antes de marcar a "festa" de renumeração. Dessa maneira, dados ultrapassados não serão colocados em cache por muito tempo e seus novos dados DNS se propagarão mais rapidamente. (Não se esqueça de modificar o TTL de volta quando tiver terminado!)

O BIND 9 impõe um conceito conhecido como harmonização TTL; ele força todos os registros num RRset (isto é, todos os registros do mesmo tipo pertencentes a um único nó) a ter o mesmo TTL. O valor que é efetivamente usado é aquele do primeiro registro de recurso para o par nó/tipo.

`$GENERATE`, uma instrução relativamente nova no BIND 8, fornece uma maneira simples de gerar uma série de registros similares. Ela serve basicamente para ajudar na geração de associações in-addr.arpa sem classes estilo RFC2317 (o truque CNAME para arquivos de zonas reversas), para casos no qual os limites da autoridade administrativa não concordem com os limites dos bytes num endereço IP.

O formato da diretiva `$GENERATE` é

```
$GENERATE início-fim/[passo] lhs tipo rhs [comentário]
```

e as linhas geradas são na forma

```
lhs tipo rhs
```

Os campos *início* e *fim* especificam o intervalo de valores para um único iterador numérico. Uma linha é gerada para cada valor no intervalo. O valor do iterador é incorporado em *lhs* e *rhs* com o caractere `$`. Se você também especificar um passo (*passo*), a iteração se dará em incrementos equivalentes a este *passo*. *tipo* é o tipo de registro. No

Snow666

momento, somente CNAME, PTR e NS são suportados no BIND 8, e esses tipos mais DNAME, A e AAAA são suportados no BIND 9. Veja Registros LOC, anteriormente neste capítulo, para ter um exemplo.

A zona de localhost

O endereço 127.0.0.1 se refere ao próprio host e sempre deve ser associado ao nome "localhost.*dominiolocal*.", por exemplo, localhost.cs.colorado.edu. Alguns sites associam o endereço simplesmente a "localhost." como se ele fosse parte do domínio-raiz; essa configuração é incorreta.

Se você se esquecer de configurar a zona localhost, seu site pode acabar consultando os servidores-raiz em busca destas informações. Atualmente, os servidores-raiz estão recebendo tantas destas consultas que os operadores estão considerando a possibilidade de adicionar uma associação genérica entre localhost e 127.0.0.1 em nível-raiz.

Consulte a página 345 para obter um exemplo de uma configuração localhost completa e correta.

Registros de adesão: links entre zonas

Cada zona é auto-suficiente com seu próprio conjunto de arquivos de dados, servidores de nomes e clientes. Porém, as zonas precisam ser conectadas para formarem uma hierarquia coerente: cs.colorado.edu é uma parte de colorado.edu, e precisamos de alguma ligação DNS entre eles.

Já que as delegações DNS ocorrem somente de domínios-pai para domínios-filho, não é necessário para um servidor de nomes saber qualquer coisa a respeito dos domínios (ou, mais precisamente, sobre as zonas) acima dele na hierarquia DNS. Entretanto, os servidores de um domínio-pai têm de conhecer os endereços IP dos servidores de nomes para todos os seus subdomínios. Na realidade, *somente* os servidores de nomes conhecidos por parte da zona-pai podem ser retornados como delegados em resposta a consultas externas.

Em termos DNS, a zona-pai precisa conter os registros NS para cada zona delegada. Pelo fato de os registros NS serem escritos em termos de nomes de hosts e não de endereços IP, o servidor-pai também tem de ter uma maneira de descobrir os nomes de hosts, seja através de uma consulta DNS normal (caso isto não crie um loop de dependência), seja por meio de cópias dos registros A apropriados.

Há duas maneiras pelas quais você pode atender a essa exigência: incluindo os registros necessários ou usando zonas stub.

Através do primeiro método, você pode simplesmente incluir os registros NS e A necessários na zona-pai. Por exemplo, o arquivo de zonas colorado.edu poderia conter os seguintes registros:

```
; informações de subdomínio
cs          IN NS ns.cs.colorado.edu.
            IN NS piper.cs.colorado.edu.
            IN NS ns.xor.com.
ee          IN NS ns.ee.colorado.edu.
            IN NS ns.cs.colorado.edu.

; registros de adesão

ns.cs       IN A  128.138.243.151
piper.cs    IN A  128.138.204.4
ns.ee       IN A  128.138.200.1
```

Os registros A "estrangeiros" são chamados de registros de adesão, pois realmente não pertencem a esta zona. Eles são reproduzidos aqui somente para conectar o novo domínio à árvore de nomes Internet. Registros de adesão faltantes ou incorretos deixarão parte de seu espaço de nomes inacessível, e os usuários que tentarem atingi-lo se depararão com erros "host unknown" (host desconhecido).

É um erro comum incluir registros de adesão para nomes de hosts que não precisem deles. Por exemplo, ns.xor.com no exemplo anterior pode ser descoberto com uma consulta DNS comum. Um registro A inicialmente seria desnecessário, porém mais tarde ele poderia se tornar totalmente enganoso caso o endereço de ns.xor.com mudasse. A regra prática é incluir registros A somente para hosts que se encontrem dentro do domínio atual ou qualquer um de seus subdomínios. As versões atuais do BIND 8 e 9 ignoram registros de adesão desnecessários e registram em log a presença deles na forma de erro.

O esquema recém-descrito é a maneira-padrão de se conectar zonas, mas ela requer que o filho permaneça em contato com o pai e informe o pai sobre quaisquer alterações ou acréscimos ao seu conjunto de servidores de nomes. Já que as zonas pai e filho normalmente são mantidas por sites diferentes, as atualizações normalmente se transfor-

www.snow666.blogspot.com
Snow666

mam numa operação manual tediosa que requer coordenação entre os diversos níveis administrativos. Um corolário é que no mundo real, este tipo de configuração normalmente se encontra desatualizado.

A segunda maneira de se fazer a manutenção de links é empregar as zonas stub. As zonas stub são totalmente suportadas no BIND 8, mas também já se encontravam disponíveis no BIND 4 (muito embora sua documentação diga que se trata de algo experimental). Uma zona stub é basicamente a mesma coisa que uma zona escrava, porém ela inclui somente os registros NS da zona.

As zonas stub funcionam bem no BIND 8, que mescla dados de zonas diferentes na memória, mas elas não funcionam tão bem com o BIND 9. No BIND 9, as zonas stub têm que ser configuradas de forma idêntica tanto nos servidores mestres quanto escravos do pai, algo que por si só é difícil de se manter consistente. O melhor que você pode fazer é simplesmente ficar em contato com o seu domínio-pai e verificar sua configuração pelo menos algumas vezes ao ano.

Você pode usar o comando **dig** para verificar quais de seus servidores seu domínio-pai está anunciando atualmente. Primeiramente execute

```
dig domínio-pai ns
```

para determinar os servidores de nomes de seu domínio-pai. Escolha um e execute

```
dig @servidor-de-nomes.domínio-pai domínio-filho ns
```

para ver a sua lista de servidores de nomes públicos. Eis um exemplo real da saída gerada por **dig** onde parte dela foi eliminada para fins de concisão:

```
% dig colorado.edu ns
;; ...
;; SEÇÃO DE RESPOSTA:
colorado.edu.          5h9m22s IN NS   ns1.westnet.net.
colorado.edu.          5h9m22s IN NS   rs0.netsol.com.
colorado.edu.          5h9m22s IN NS   boulder.colorado.edu.
colorado.edu.          5h9m22s IN NS   cujo.colorado.edu.
% dig @boulder.colorado.edu cs.colorado.edu ns
;;; SEÇÃO DE RESPOSTA:
cs.colorado.edu.       6H IN NS        cs.colorado.edu.
cs.colorado.edu.       6H IN NS        huizil.cs.colorado.edu.
cs.colorado.edu.       6H IN NS        xor.com.
cs.colorado.edu.       6H IN NS        pacifier.com.
```

Somente quatro servidores para o domínio cs.colorado.edu são visíveis do mundo externo.

Um comando **dig** realizado de dentro do departamento resultaria numa lista diferente:

```
;; SEÇÃO DE RESPOSTA:
cs.colorado.edu.       2H IN NS        cs.colorado.edu.
cs.colorado.edu.       2H IN NS        moet.cs.colorado.edu.
cs.colorado.edu.       2H IN NS        piper.cs.colorado.edu.
cs.colorado.edu.       2H IN NS        anchor.cs.colorado.edu.
cs.colorado.edu.       2H IN NS        vulture.cs.colorado.edu.
```

Apenas o servidor cs.colorado.edu principal é utilizado tanto dentro quanto fora do departamento.

Uma situação na qual as zonas stub são muito úteis é aquela em que seu endereçamento interno usa o espaço de endereços privado RFC1918 e você precisa manter as delegações RFC1918 em sincronia. A distribuição BIND 8 contém um exemplo em **src/conf/recursive**.

Uma série de sutilezas de zonas stub que valem a pena mencionar são as seguintes:

- As zonas stub não são cópias autorizadas dos dados da zona, e os servidores stub não devem ser listados entre os registros NS da zona.
- Já que os servidores stub não são listados em registros NS, eles não são notificados automaticamente quando os dados da zona mudam. Para atualizar servidores stub, podemos adicionar uma instrução `also-notify` à configuração dos servidores mestres ou então simplesmente aguardar que a zona seja atualizada no final do intervalo de atualização especificado no registro SOA da zona. A opção `timeout` deve funcionar perfeitamente na maioria dos casos, entretanto há a possibilidade de resultar em delegações transitórias imperfeitas (veja a página 343).

www.snow666.blogspot.com
Snow666

- Teoricamente, não tem valor algum para **named** ter cópias dos registros NS de uma zona se ele também não puder obter os registros A concordantes. Entretanto, **named** é capaz de se virar por conta própria usando o endereço IP do mestre, que é dado em **named.conf**.
- Por que se limitar somente aos registros NS? Por que não simplesmente ser um servidor secundário para os subdomínios? Isso também funciona. Entretanto, se cada servidor do domínio-pai também for um servidor de um domínio-filho, jamais será feita alguma delegação a servidores abaixo. Os servidores do domínio-pai fornecerão todo o serviço DNS para o subdomínio. Talvez isso seja o que você deseja, talvez não.

16.12 Atualizando Arquivos de Zonas

Ao efetuar uma alteração num domínio (como acrescentar ou eliminar um host), os arquivos de dados no servidor mestre devem ser atualizados. Você também tem de incrementar o número de série no registro SOA para a zona e depois executar **ndc reload** (**rndc reload** no BIND 9) para sinalizar para **named** que colete as modificações. Você também pode matar e reiniciar o processo **named** (**ndc restart**), porém essa operação faz com que dados colocados em cache provenientes de outros domínios sejam descartados.

Versões mais antigas do BIND usavam sinais e o comando **kill** do Linux para controlar **named**, mas somente quando os desenvolvedores começaram a ficar sem números de sinalização, o **ndc** entrou em cena e corrigiu tudo. A maior parte da sinalização histórica no BIND (exceto pelo sinal HUP para reler o arquivo de configuração e o sinal TERM para terminar) provavelmente desaparecerá em versões futuras; portanto, recomendamos que você use **ndc**.

Os dados de zonas atualizadas são propagados para servidores escravos imediatamente, pois a opção `notify` fica ativa como padrão. Se, inadvertidamente, você desabilitar essa opção, seus servidores escravos não coletarão as alterações antes de serem decorridos *atualizar* segundos, conforme configurado no registro SOA da zona (tipicamente de uma a seis horas depois). Se você quiser uma atualização mais oportuna quando a opção `notify` estiver desabilitada, o **ndc reload** num escravo faz com que ele verifique com o mestre, veja que dados foram modificados e solicite uma transferência de zonas.

Não se esqueça de modificar tanto as zonas diretas quanto as reversas ao modificar o nome de um host ou endereço IP. Esquecer dos arquivos reversos deixa erros sorrateiros: alguns comandos funcionam e outros não.

Alterar os arquivos de dados mas se esquecer de alterar o número de série faz com que as alterações entrem em efeito no servidor mestre (após uma recarga), mas não nos escravos.

É inadequado editar arquivos de dados pertencentes aos servidores escravos. Esses arquivos são mantidos por **named**; os administradores de sistemas não devem se meter com eles. Não há nenhum problema em verificar os arquivos de dados desde que você não faça alterações.

O BIND permite que sejam realizadas alterações em zonas por meio de uma API (Application Program Interface, interface de programação de aplicação) de programação, conforme especificado na RFC2136. Este recurso, denominado atualização dinâmica, é necessário para protocolos de autoconfiguração como o DHCP. O mecanismo de atualização dinâmica é descrito mais adiante.

Transferências de zonas

Os servidores DNS são sincronizados por um mecanismo chamado transferência de zonas. A especificação DNS original (e BIND 4) requeria que todos os dados de zona fossem transferidos de uma só vez. Atualizações incrementais foram por fim definidas na RFC1995 e implementadas no BIND 8.2. Transferências de zonas originais e do tipo incremental são algumas vezes chamadas de, respectivamente, AXFR e IXFR. Uma vez configuradas, supostamente elas são equivalentes.

Um escravo que quer atualizar seus dados solicita uma transferência de zona do servidor mestre e faz uma cópia backup dos dados de zona em disco. Se os dados no mestre não tiverem sido modificados, conforme determinado por uma comparação dos números de série (não os dados em si), não ocorrerá nenhuma atualização de dados e os arquivos de backup serão somente tocados (isto é, o horário de modificação deles será configurado para o horário atual).

As transferências de zonas usam o protocolo TCP na porta 53 e registram informações em log através de syslog com a marca "named-xfer". IXFR, conforme especificado pelo IETF, pode usar tanto TCP quanto UDP, porém BIND somente o implementou através de TCP.

Tanto o servidor que está enviando quanto o que está recebendo permanecem disponíveis para responder a consultas durante uma transferência de zona. Somente depois de a transferência ter sido completada o escravo começa a usar os novos dados. O BIND 8 na verdade chama um programa **named-xfer** separado para realizar a transferência, porém o processo **named** do BIND 9 trata diretamente das transferências. Portanto, a opção `named-xfer` que especificava o caminho para o programa **named-xfer** não faz mais parte da linguagem de configuração do BIND 9.

Snow666

Quando as zonas são imensas (como com) ou atualizadas dinamicamente (veja a próxima seção), as alterações são tipicamente menores em relação ao tamanho de toda a zona. Com o IXFR, somente as alterações são enviadas (a menos que elas sejam maiores do que a zona inteira, em cujo caso é feita uma transferência AXFR comum). O mecanismo IXFR é como o programa **patch** no aspecto em que ele aplica diferenças a um banco de dados antigo para fazer com que entre em sincronismo com um novo banco de dados.

No BIND 8, habilitamos o IXFR informando **named** para manter um log de transações na seção `options` global e depois o ativando nas instruções de servidor para quaisquer servidores que fizerem uso deste. As linhas relevantes do arquivo de configuração são:

```
maintain-ixfr-base true ;          # na seção de opções do BIND 8
use-ixfr true ;                    # na instrução server do BIND 8
```

Caso queira alterar os nomes-padrão para o log de transações e arquivo temporário usado por IXFR, faça-o nas instruções `zone` através de:

```
ixfr-base "nome-de-arquivo" ;      # em instruções zone do BIND 8
ixfr-tmp-file "nome-de-arquivo" ;  # em instruções zone do BIND 8
```

No BIND 9, IXFR é o padrão para qualquer zona configurada para atualização dinâmica e **named** sempre mantém um log de transações chamado *nome-da-zona*.**jnl**. Você pode configurar as opções `provide-ixfr` e `request-ixfr` nas instruções `server` para peers individuais. `provide-ixfr` habilita ou desabilita o serviço IXFR para zonas para as quais este servidor é o mestre. `request-ixfr` `solicita` IXFRs para zonas para as quais este servidor é um escravo.

```
provide-ixfr yes ;                 # na instrução server do BIND 9
request-ixfr yes ;                 # na instrução server do BIND 9
```

O BIND não é capaz de lidar com uma zona que é atualizada tanto de maneira dinâmica quanto editada manualmente. O BIND 9 fornece IXFR de saída para alterações resultantes de atualizações dinâmicas ou de IXFRs que chegaram, mas não para alterações resultantes de edições nos arquivos de zona mestra. Esse recurso provavelmente será adicionado numa versão futura.

Foi investido muito trabalho no mecanismo IXFR para garantir que um travamento no servidor durante uma atualização não deixasse as zonas com dados desnecessários ou sem significado. Uma solicitação IXFR a um servidor que não o suporte automaticamente retorna a uma transferência de zona AXFR padrão.

Atualizações dinâmicas

O sistema DNS se baseia na premissa de que as associações nome-endereço são relativamente estáveis e não são alteradas freqüentemente. Porém, um site que usa DHCP para atribuir endereços IP dinamicamente enquanto as máquinas são inicializadas e se conectam à rede quebra essa regra constantemente. Há duas soluções clássicas: adicionar entradas genéricas ao banco de dados DNS ou editar continuamente os arquivos DNS. Para muitos sites, nenhuma das soluções é satisfatória.

A primeira solução deve ser familiar para qualquer um que tenha usado um ISP discado. A configuração DNS ficará parecida como o seguinte:

```
dhcp-host1.domain.    IN A  192.168.0.1
dhcp-host2.domain.    IN A  192.168.0.2
...
```

Embora essa seja uma solução simples, isso significa que nomes de hosts são associados permanentemente a determinados endereços IP e que os computadores, conseqüentemente, mudam nomes de hosts sempre que eles receberem um novo endereço IP. Registro baseado em nomes de hosts ou medidas de segurança se tornam muito difíceis nessa situação.

O recurso de atualização dinâmica em versões recentes do BIND fornece uma solução alternativa. Ele permite que o daemon DHCP notifique o BIND sobre as atribuições de endereços feitas atualizando, conseqüentemente, o conteúdo do banco de dados DNS em tempo real. Uma interface de shell também é fornecida para se realizar atualizações dinâmicas manualmente.

As atualizações dinâmicas podem adicionar, eliminar ou modificar registros de recursos. A granularidade na qual as atualizações dinâmicas são reguladas é a zona. É um tanto temeroso permitir atualizações dinâmicas no banco de dados DNS completo de seu site, de modo que muitos sites criam um subdomínio (talvez dhcp.*site*) e permitem atualizações dinâmicas somente dentro deste subdomínio.

Snow666

Assim que uma zona tiver sido atualizada dinamicamente, você não poderá editá-la manualmente sem primeiro interromper o BIND com o comando **ndc stop** (**rndc stop** no BIND 9) para que a cópia atual do banco de dados possa ser gravada no disco. Você poderá então editar manualmente o arquivo de zonas. Obviamente, a formatação original do arquivo de zonas será destruída (o arquivo ficará parecido com aquele mantido por **named** para servidores escravos).

Enquanto as atualizações dinâmicas estão ocorrendo, um arquivo com registro de ações (*journal file*) (*nome-da-zona*.**jnl**) é mantido no caso de um travamento no servidor. Esse arquivo tem de ser eliminado após você interromper **named** e antes de começar a editar o arquivo de zonas manualmente. Entretanto, realmente não é bom se meter com zonas DNS dinâmicas, pois poderão resultar em dados inconsistentes.

As transferências de zonas incrementais são o padrão para zonas que usam atualizações dinâmicas. Elas utilizam o mesmo mecanismo de registro de ações que as atualizações dinâmicas.

O programa **nsupdate** fornecido juntamente com o BIND 9 fornece uma interface de linha de comando para atualizações dinâmicas. Ela é executada em modo lote, extraindo comandos do teclado ou de um arquivo. Duas linhas em branco sinalizam o final da entrada. A linguagem de comando inclui uma instrução primitiva "if" para expressar construções do tipo: "se este nome de host não existir no DNS, acrescente-o". Como predicados para uma ação **nsupdate**, você pode requerer que um nome exista ou não, ou requerer que um conjunto de registros de recursos exista ou não.

Eis, por exemplo, um script **nsupdate** simples que acrescenta um novo host e também adiciona um apelido para um host existente se o apelido já não estiver em uso:

```
% nsupdate
> update add newhost.cs.colorado.edu 86400 A 128.138.243.16
>
> prereq nxdomain gypsy.cs.colorado.edu
> update add gypsy.cs.colorado.edu CNAME evi-laptop.cs.colorado.edu
```

Realizar atualizações dinâmicas no DNS é amedrontador. Elas têm o potencial de fornecer acesso de gravação incontrolado para dados de sistemas importantes. Não tente usar endereços IP para controle de acesso — eles são facilmente forjados. A autenticação TSIG com uma chave secreta compartilhada é mais indicada; ela se encontra disponível e é fácil de ser configurada. Ambas as versões 8 e 9 do BIND a suportam

```
% nsupdate -k diret rio-da-chave:arquivo-da-chave
```

e o BIND 9 também entende

```
% nsupdate -y nome-da-chave:chave-secreta
```

Para maiores detalhes sobre o TSIG, consulte a seção que começa na página 328.

As atualizações dinâmicas para uma zona são habilitadas em **named.conf** através de uma cláusula `allow-update` ou `update-policy`. `allow-update` concede permissão para atualizar quaisquer registros de acordo com o endereço IP de origem ou autenticação baseada em chave. `update-policy` é uma extensão do BIND 9 que permite controle fino para atualizações de acordo com o nome de host ou tipo de registro. Ela requer autenticação baseada em chave.

Use `update-policy` para permitir que os clientes atualizem seus registros A ou PTR, mas não alterem o registro SOA, registros NS ou registros KEY. Você também pode usá-la para permitir que um host atualize somente seus próprios registros. Os parâmetros permitem que você expresse nomes explicitamente, na forma de um subdomínio, na forma de um curinga ou na forma da palavra-chave `key`, que configura uma política genérica para acesso das máquinas a seus próprios registros. Os registros de acesso são identificados por classe e tipo. Eis um exemplo:

```
update-policy { grant chave-dhcp subdomínio dhcp.cs.colorado.edu A } ;
```

Essa configuração permite que qualquer um que conheça o código `chave-dhcp` consiga atualizar registros de endereços no subdomínio dhcp.cs.colorado.edu. Essa instrução apareceria no arquivo **named.conf** sob a instrução `zone` para dhcp.cs.colorado.edu. Deveria ter uma instrução `key` para definir `chave-dhcp` também. Veja a documentação do BIND para obter todos os detalhes referentes à instrução `update-policy`.

16.13 Questões de Segurança

Duas coisas mantêm a Internet coesa: o sistema DNS e o sistema de roteamento. Nos velhos tempos, a Internet era pequena, amigável e usada em grande parte por geeks. Hoje ela é um ambiente hostil, bem como uma parte crucial de infra-estrutura. Esta seção discute vários tópicos relacionados à segurança de uma maneira que pode parecer draconiana e paranóica. Infelizmente, estes tópicos e precauções são importantes e tristemente necessários na Internet atual.

Snow666

O DNS começou como um sistema inerentemente aberto, porém ele foi crescendo de forma mais segura — ou pelo menos, em condições de tornar-se seguro. Como padrão, qualquer um na Internet pode investigar seu domínio com consultas individuais usando ferramentas como **dig**, **host** ou **nslookup**. Em alguns casos, elas podem fazer um dump completo de seu banco de dados DNS.

Para contornar essas vulnerabilidades, o BIND agora suporta vários tipos de controle de acesso baseado em endereços de host e de rede ou em autenticação criptografada. A Tabela 16.11 sintetiza os recursos de segurança que são configurados em **named.conf**. A coluna Página mostra onde procurar informações adicionais neste capítulo.

Tabela 16.11 Recursos de segurança em named.conf.

Recurso	Instruções	Página	O que ele especifica
allow-query	options, zone	291	Quem pode consultar uma zona ou servidor
allow-transfer	options, zone	291	Quem pode solicitar transferências de zonas
allow-update	zone	293	Quem pode realizar atualizações dinâmicas
update-policy	zone	325	Que atualizações são permitidas
blackhole	options	291	Que servidores ignorar completamente
bogus	server	292	Que servidores jamais devem ser consultados
acl	várias	291	Listas de controle de acesso

named é capaz de ser executado num ambiente em que foi aplicado **chroot** sob um UID não privilegiado para minimizar riscos de segurança. Ele pode usar assinaturas de transação para controlar atualizações dinâmicas e, obviamente, também suporta todo o emaranhado do DNSSEC. Esses tópicos são vistos nas próximas seções.

Retornando às listas de controle de acesso

As ACLs são listas de concordância de endereços com nomes que podem aparecer como argumentos em instruções como `allow-query`, `allow-transfer` e `blackhole`. As ACLs podem auxiliar em duas questões importantes de segurança relacionadas com o DNS: ataques de falsificação (spoofing) e ataques de negação de serviços. Suas sintaxes básicas foram descritas na seção 16.9.

Cada site deveria ter ao menos uma ACL para endereços não genuínos e uma ACL para endereços locais. Por exemplo:

```
acl bogusnets {                 // ACL para redes não genuínas
      0.0.0.0/8 ;               // padrão, endereços-curinga
      1.0.0.0/8 ;               // endereços reservados
      2.0.0.0/8 ;               // endereços reservados
      169.254.0.0/16 ;          // endereços link-local delegados[16]
      192.0.2.0/24 ;         // amostras de endereços, como example.com
      224.0.0.0/3 ;          // espaço de endereços multidifusão
      10.0.0.0/8 ;           // espaço de endereços privado (RFC1918)[17]
      172.16.0.0/12 ;        // espaço de endereços privado (RFC1918)
      192.168.0.0/16 ;       // espaço de endereços privado (RFC1918)
} ;

acl cunets {                 // ACL para as redes da Universidade do Colorado
      128.138.0.0/16 ;       // rede principal do campus
      198.11.16/24 ;
      204.228.69/24 ;
};
```

Na seção `options` global de seu arquivo de configuração, você poderia então incluir

```
allow-recursion { cunets; } ;
```

16. O endereço link-local é usado por Macs e PCs que foram informados para utilizarem IP, mas não conseguem localizar um servidor DHCP. Essas máquinas simplesmente atribuem a elas próprias um endereço na rede 169.254.0.0/16. Endereços nesse intervalo devem ser filtrados agressivamente para que jamais escapem da rede local. Modems a cabo ou DSL também estão começando a usar esse intervalo de endereços.
17. Não tranforme endereços privados em não genuínos caso você os utilize e esteja configurando os seus servidores DNS internos.

Snow666


```
blackhole { bogusnets; } ;
```

Também é uma boa idéia restringir transferências de zonas para legitimar servidores escravos. Uma ACL facilita e organiza as coisas.

```
acl ourslaves {
      128.138.242.1 ;        // anchor
      …
} ;

acl measurements {
      128.9.160.157 ;        // medidas de bill manning
      198.32.4.0/24 ;        // medidas de bill manning
      192.5.5.0/24 ;         // medidas de mark lottors
} ;
```

A real restrição é implementada por uma linha como:

```
allow-transfer { ourslaves; measurements; } ;
```

As transferências são limitadas aos nossos próprios servidores escravos e às máquinas de dois projetos de medidas Internet que percorrem a árvore DNS ao contrário para determinar o tamanho da Internet e a porcentagem de servidores mal configurados. Limitar transferências dessa maneira torna impossível para outros sites fazerem um dump completo de seu banco de dados através de **nslookup**, **dig** ou **host**.

Por exemplo, usando **dig @***servidor domínio-a-descarregar* **axfr**:

```
% dig @128.138.143.151 cs.colorado.edu axfr
;; connect: Operation timed out

; <<>> DiG 8.3 <<>> @128.138.143.151 cs.colorado.edu axfr
; (1 server found)
;; FROM: chateau.caida.org to SERVER: 128.138.143.151
;; WHEN: Tue Feb 19 17:16:53 2002
```

Você ainda pode proteger sua rede em um nível mais baixo através de listas de controle de acesso de roteador e medidas de higienização de segurança normais em cada host. Se essas medidas não forem possíveis, você poderá recusar pacotes DNS, exceto para uma máquina de gateway que você monitora de perto.

Confinando named

Para confinar o dano que alguém poderia causar caso comprometesse seu servidor, você poderá executar **named** num ambiente onde foi aplicado **chroot** e/ou executá-lo como usuário não privilegiado. O flag **-t** especifica o diretório para onde se deve fazer o **chroot**, e os flags **-u** e **-g** especificam o UID e GID sob os quais deve ser feita a execução. O BIND 9 suporta o flag **-u**, mas não o flag **-g**. Por exemplo, os comandos

```
# named -u 53 -g 53 -t /var/named     /* BIND 8 */
# named -u 53 -t /var/named           /* BIND 9 */
```

iniciam **named** com UID 53, GID 53 (BIND 8), e um diretório raiz **/var/named**.

O diretório **chroot** não pode ser um diretório vazio já que tem de conter todos os arquivos que **named** normalmente requer para ser executado: **/dev/null**, bibliotecas compartilhadas, arquivos de zona, **named.conf**, **named-xfer**, arquivos de destino de syslog e socket de domínio UNIX, **/var**, etc. Se você puder compilar **named** para linkar estaticamente suas bibliotecas, não terá de descobrir que arquivos de biblioteca copiar para **/var/named**. Veja o site do ISC http://www.isc.org/products/BIND/contributions.html com links que fornecem documentos com informações úteis.

O livro DNS da O'Reilly também fornece instruções similares. O SuSE possui sua própria página "como fazer" sobre ambientes em que se aplica **chroot**

```
/usr/share/doc/howto/en/html/Chroot-B ND- O TO.html
```

O BIND 9 é mais fácil de ser configurado para executar ambientes em que se aplicou **chroot**, pois **named-xfer** é embutido e a chamada de sistema **chroot** é feita depois de todas as bibliotecas terem sido carregadas.

Se hackers comprometerem seu processo **named**, eles poderão ganhar acesso ao sistema como qualquer usuário em que **named** é executado. Se esse usuário for root e você não usar um ambiente no qual se aplicou **chroot**, tal

www.snow666.blogspot.com
Snow666

brecha pode ser bastante destrutiva. Muitos sites não se preocupam em usar os flags **-u**, **-g** e **-t**, mas depois eles têm de ser mais rápidos na atualização do que os hackers em termos de ataque quando for anunciada uma nova vulnerabilidade.

Comunicação servidor-servidor segura com TSIG e TKEY

Enquanto o DNSSEC (coberto na próxima seção) estava sendo especificado, o IETF desenvolveu um mecanismo mais simples chamado TSIG (RFC2845) para possibilitar comunicação segura entre servidores por meio do emprego de assinaturas de transação. O controle de acesso baseado em assinaturas de transação é mais seguro do que o controle de acesso baseado em endereços de origem IP. A assinatura de transação autentica o emissor/receptor e verifica se os dados não foram adulterados. Ela é usada tipicamente para transferências seguras de zonas entre um mestre e seus escravos e para atualizações dinâmicas seguras.

As assinaturas de transação usam um esquema de criptografia simétrica. Ou seja, a chave de criptografia é a mesma que a chave de decifrar. Essa única chave é chamada de chave de segredo compartilhado. Você deve usar uma chave diferente para cada par de servidores com os quais deseja se comunicar de forma segura. O TSIG é muito menos caro computacionalmente do que a criptografia de chaves públicas, mas ele é apropriado somente para uma rede local na qual o número de pares de servidores de comunicação é pequeno. Ele não pode ser expandido para a Internet global.

As assinaturas TSIG assinam consultas DNS e respostas a consultas. Elas são usadas somente entre servidores, não entre servidores e solucionadores. As assinaturas TSIG são verificadas no momento em que um pacote é recebido e são depois descartadas; elas não são colocadas em cache e não passam a fazer parte dos dados DNS. Embora a especificação TSIG permita múltiplos métodos de criptografia, o BIND implementa somente um, o algoritmo HMAC-MD5.

O utilitário **dnssec-keygen**[18] do BIND gera uma chave para um par de servidores. Por exemplo, para gerar uma chave host de segredo compartilhado para dois servidores, *serv1* e *serv2*, use

```
# dnssec-keygen -a  MAC-MD5 -b 128 -n  OST serv1-serv2
```

para criar uma chave de 128 bits. São gerados dois arquivos: **K***serv1-serv2***.+157+09068.private** e **K***serv1-serv2***.+157+09068.key**. **157** significa o algoritmo HMAC-MD5; **09068** é um número aleatório usado como identificador de chaves caso você tenha várias chaves para o mesmo par de servidores. O arquivo **.private** fica parecido com o seguinte:

```
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: jxopbeb+aPc71Mm2vc9R9g==
```

e o arquivo **.key** se parece com:

```
serv1-serv2. IN KEY 512 3 157 jxopbeb+aPc71Mm2vc9R9g==
```

Ambos os arquivos devem ter o modo 600 e devem pertencer ao usuário **named**.[19] Na verdade, você não precisa do arquivo **.key** em si — ele é produzido porque o programa **dnssec-keygen** também é usado para gerar pares de chaves públicas na qual a chave pública (arquivo **.key**) é inserida no arquivo de zonas DNS como um registro de recursos KEY.

A chave gerada é realmente apenas um longo número aleatório. Você poderia gerar a chave manualmente escrevendo uma string ASCII do comprimento exato e fazendo com que esta se passasse por uma codificação de base 64 de alguma coisa ou usando **mmencode** para codificar uma string aleatória. (**mmencode** se encontra na instalação-padrão das distribuições Red Hat e SuSE, mas você tem de aplicar **apt-get** no Debian.) A maneira pela qual você cria a chave não é importante; ela apenas tem de existir em ambas as máquinas.

scp faz parte da família SSH. Consulte a seção 21.9 para maiores detalhes.

Copie a chave tanto para *serv1* quanto para *serv2* usando **scp** ou então recortando-a e colando-a. *Não* use **telnet** ou **ftp** para copiar a chave; mesmo redes internas podem não ser seguras. A chave tem que ser incluída nos arquivos **named.conf** de ambas as máquinas. Já que **named.conf** normalmente pode ser lido por todo o mundo e as chaves não, coloque a chave num arquivo distinto que é incluso em **named.conf**. Você poderia, por exemplo, colocar o pequeno trecho de código a seguir

18. Esse comando chama-se **dnskeygen** no BIND 8.
19. O comando **nsupdate** do BIND 8 falhará silenciosamente (nenhuma indicação, nenhuma mensagem de erro) caso o arquivo **.key** não seja legível.

Snow666


```
key serv1-serv2 {
   algorithm hmac-md5 ;
   secret "chave-secreta-que-foi-gerada" ;
} ;
```

no arquivo **serv1-serv2.key**. O arquivo deve ter o modo 600 e seu proprietário deve ser o UID de **named**. No arquivo **named.conf**, acrescente a seguinte linha

```
include "serv1-serv2.key"
```

próximo do início do arquivo.

Esta parte da configuração simplesmente define as chaves. Para fazer com que elas sejam efetivamente usadas para assinar e verificar atualizações, cada servidor precisa identificar o outro com uma cláusula `keys`.

Você poderia, por exemplo, acrescentar as linhas

```
server serv2'endereço-IP {
   keys { serv1-serv2 ; } ;
} ;
```

ao arquivo **named.conf** de serv1 e

```
server serv1'endereço-IP {
   keys { serv1-serv2 ; } ;
} ;
```

ao arquivo **named.conf** de serv2. Qualquer cláusula `allow-query`, `allow-transfer` e `allow-update` na instrução `zone` para a zona também deveria se referir à chave. Por exemplo:

```
allow-transfer { key serv1-serv2 ;} ;
```

Ao começar a usar assinaturas de transação, execute **named** no nível de depuração 1 (veja a página 335 para obter informações sobre como executar **named** no modo de depuração) por algum tempo para ver quaisquer mensagens de erro geradas. Versões mais antigas de BIND não compreendem mensagens assinadas e reclamam delas, algumas vezes chegando a ponto de se recusarem a carregar a zona.

TKEY é um mecanismo do BIND 9 que permite que dois hosts gerem uma chave de segredo compartilhado automaticamente sem chamadas telefônicas ou cópias seguras para distribuir a chave. Ele usa um algoritmo chamado intercâmbio de chaves Diffie-Hellman no qual cada lado forma um número aleatório, aplica o mesmo algoritmo sobre ele e envia o resultado para o outro lado. A seguir, cada lado combina matematicamente seu próprio número com a transmissão que ele recebeu para chegar à mesma chave. Alguém na escuta poderia capturar a transmissão, mas seria incapaz de reverter os cálculos efetuados.[20]

DNSSEC

DNSSEC é um conjunto de extensões DNS que autenticam a origem dos dados de zonas e verificam sua integridade usando criptografia de chaves públicas. Isto é, as extensões permitem que clientes DNS façam as seguintes perguntas: "Estes dados DNS realmente provêm do proprietário da zona?" e "Estes são realmente os dados enviados por este proprietário?".

O DNSSEC fornece três serviços distintos: distribuição de chaves por meio de registros de recursos KEY armazenados nos arquivos de zonas, verificação de origem para servidores e dados e verificação da integridade dos dados da zona. O DNSSEC é baseado em uma cadeia em cascata de confiabilidade: os servidores-raiz fornecem informações de validação para os domínios de primeiro nível, os domínios de primeiro nível fornecem informações de validação para os domínios de segundo nível e assim por diante.

Sistemas de criptografia de chaves públicas usam duas chaves: uma para criptografar (assinar) e uma outra diferente para decifrar (verificar). Os publicadores assinam seus dados com uma chave secreta "privada". Qualquer um pode verificar a validade de uma assinatura com uma chave "pública" concordante que é amplamente distribuída. Se uma chave pública decifra corretamente um arquivo de zonas, então a zona só pode ter sido criptografada com a chave privada correspondente. O truque é ter certeza de que as chaves públicas que você usa para verificação são autênticas. Os sistemas de chaves públicas permitem que uma entidade assine a chave pública de uma outra atestando assim a legitimidade da chave; daí o termo "cadeia de confiabilidade".

20. Os cálculos envolvidos são chamados de problema de logaritmo distinto e depende do fato de que, para aritmética modular, calcular potências é fácil, mas calcular logaritmos para desfazer as potências é quase impossível.

Snow666

Os dados numa zona DNS são muito volumosos para serem cifrados com criptografia de chaves públicas — a criptografia seria muito lenta. Em vez disso, já que os dados não são secretos, um hash[21] seguro (por exemplo, um checksum MD5) é executado nos dados e os resultados do hash são assinados (cifrados) pela chave privada da zona. Os resultados do hash são equivalentes a uma impressão digital dos dados. A impressão digital assinada é chamada de assinatura digital.

As assinaturas digitais normalmente são anexadas aos dados que elas autenticam. Para verificar a assinatura, você a decifra com a chave pública do assinante, passa os dados pelo mesmo algoritmo hash seguro e compara o valor hash calculado com o valor hash decifrado. Se eles forem coincidentes, você autenticou o assinante e verificou a integridade dos dados.

No sistema DNSSEC, cada zona possui suas próprias chaves pública e privada. A chave privada assina cada RRset (ou seja, cada conjunto de registros do mesmo tipo para o mesmo host). A chave pública verifica as assinaturas e é incluída nos dados da zona na forma de um registro de recursos KEY.

Zonas-pai assinam as chaves públicas de suas zonas-filha. **named** verifica a autenticidade do registro KEY de uma zona-filha comparando-a com a assinatura da zona-pai. Para verificar a autenticidade da chave da zona-pai, **named** é capaz de verificar o pai do pai e assim por diante até chegar à raiz. A chave pública para a zona-raiz seria incluída no arquivo de dicas raiz.

Antes de mergulharmos nos mecanismos de geração de chaves e de assinatura de zonas precisamos ser honestos em relação ao estado atual do DNSSEC e seu impacto sobre administradores de sistemas. Muitos aplicativos estão clamando uma infra-estrutura de chaves públicas ("PKI" - "ICP"), e o DNS é um candidato em potencial para fornecê-las. Entretanto, temos um pouco daquele problema do ovo e da galinha. Precisamos ter certeza de que o DNS é seguro antes de podermos confiar a ele nossas chaves para outras transações na Internet . Porém, precisamos de uma infra-estrutura de chaves públicas para assegurar o DNS.

Sistemas como o PGP (Pretty Good Privacy de Phil Zimmermann) dependem de um conceito de "malha de confiabilidade". Você acreditará que algo é minha chave se alguém em quem você confia disser isto. A idéia no DNSSEC é a de criar uma malha de confiabilidade da zona-pai para a zona-filha assinando chaves e colocando as assinaturas nos dados DNS. Precisamos apenas disparar o processo e possibilitar o emprego parcial.

Administradores de sistemas precisarão lidar com DNSSEC dentro de um a dois anos, mas certamente ele não se encontra na lista de obrigações desta semana. Chega de política, retornemos à mecânica de geração de chaves e de assinatura de suas zonas ...

São necessárias várias etapas para criar e usar zonas com assinatura. Em primeiro lugar, você gera um par de chaves para a zona. Por exemplo, no BIND 9,

```
# dnssec-keygen -a DSA -b 768 -n  ONE meudominio.com
```

ou no BIND 8,

```
# dnskeygen -D768 -z -n meudominio.com
```

A Tabela 16.12 ilustra o significado dos argumentos para esses comandos.

Tabela 16.12 Guia de decodificação para argumentos dnssec-keygen e dnskeygen.

Para o dnssec-keygen	
-a DSA	Usa o algoritmo DSA
-b 768	Cria um par de chaves de 768 bits
-n Zone meudominio.com	Cria chaves para uma zona chamada
Para o dnskeygen	
-D768	Usa o algoritmo DSA, com uma chave de 768 bits
-z	Cria uma chave de zona
-n meudominio.com	Cria chaves para uma zona chamada meudominio.com

dnssec-keygen e **dnskeygen** retornam a seguinte saída:

21. Hash é uma operação matemática aplicada a um conjunto de dados de entrada, produzindo uma seqüência finita de caracteres que depende de todos os bits daquele conjunto de dados. Qualquer mudança, por menor que seja, no conjunto de dados produziria outra seqüência de caracteres. Daí sua aplicação à segurança de sistemas. (N.R.T.)

Snow666


```
alg = 003
key identifier = 12345            # uma string aleatória de ID da chave
flags = 16641
```

Eles também criam arquivos contendo as chaves públicas e privadas:

```
Kmeudominio.com.+003+12345.key        # chave pública
Kmeudominio.com.+003+12345.private    # chave privada
```

A chave privada é usada para assinar os registros de dados de uma zona e a chave pública é usada para verificar assinaturas. A chave pública é colocada tipicamente no arquivo de zonas logo após o registro SOA.

DNSSEC requer uma cadeia de confiabilidade; portanto, uma chave pública tem de ser assinada por seu pai para ser confirmadamente válida. O BIND 8 não tinha nenhum mecanismo para fazer com que uma zona-pai assinasse a chave de uma zona-filha a não ser cooperação externa entre os administradores de sistemas. O BIND 9 dispõe de um programa chamado **dnssec-makekeyset** para ajudar neste processo.

dnssec-makekeyset empacota as chaves que você quer que sejam assinadas (pode haver mais do que uma chave de zona), um TTL para o conjunto de chaves resultante e um período de validade de assinatura, depois envia o pacote para o pai para assinatura. Por exemplo, o comando

```
# dnssec-makekeyset -t 3600 -s +0 -e +864000 Kmeudominio.com.+003+12345
```

empacota a chave de zona pública que você acabou de gerar com um TTL de 3.600 segundos (uma hora) e solicita que a assinatura do pai seja válida por 10 dias a partir deste momento. **dnssec-makekeyset** cria um único arquivo de saída, **keyset.meudominio.com**. Você tem então de enviar o arquivo para a zona-pai para assinatura. Ele contém a chave pública e assinaturas geradas pelas próprias chaves de zona que o pai possa verificar a chave pública do filho.

No BIND 9, a zona-pai usa o programa **dnssec-signkey** para assinar o conjunto empacotado de chaves:

```
# dnssec-signkey keyset.meudominio.com Kcom.+003+56789
```

Esse comando gera um arquivo chamado **signedkey.meudominio.com**, o qual o pai (com) envia de volta para o filho (meudominio.com) para ser incluído nos arquivos de zonas para meudominio.com. No BIND 8, o pai usa o comando **dnssigner**.

Assim que você tiver obtido a assinatura do pai, estará pronto para assinar os dados da zona em si. A operação de assinatura usa um arquivo de dados de zonas comum como entrada e acrescenta registros SIG e NXT imediatamente após cada conjunto de registros de recursos. Os registros SIG são as assinaturas em si, e os registros NXT suportam assinatura de respostas negativas.

No BIND 8, você usa o programa **dnssigner** contido no diretório **contrib** da distribuição para assinar uma zona; no BIND 9, você usa o comando **dnssec-signzone**. Por exemplo, os comandos

```
# dnssigner -or meudominio.com -zi db.meudominio -zo
   db.meudominio.signed -k1 meudominio.com dsa 12345 -st    # BIND 8
# dnssec-signzone -o meudominio.com db.meudominio           # BIND 9
```

lêem o arquivo de zonas **db.meudominio** e produzem uma versão assinada do arquivo de zonas chamada **db.meudominio.signed**. Os comandos do BIND 8 exibem várias informações estatísticas sobre a operação de assinatura (solicitada pelo parâmetro **-st**). Em particular, eles mostram quanto tempo leva para assinar a zona e quais registros foram adicionados ou eliminados. Pode levar um bom tempo para assinar uma zona.

O processo de assinatura de zonas acrescenta registros SIG e NXT a cada conjunto de registros de recursos. O exemplo a seguir mostra os panoramas antes e depois. Eis os panoramas antes:

```
anchor      IN  A     128.138.242.1
            IN  A     128.138.243.140
            IN  MX    10 anchor
            IN  MX    99 @
awesome     IN  A     128.138.236.20
...
```

E depois:

```
$ORIGIN meudominio.com.
anchor          SIG    MX 3 3 3600 20001008023531 (20000908023531 12345
                       meudominio.com. BFEtOCT+y0dQPx7Am7gpxD9SjEl+
                       USuaE7qExUOrX22X7wjqJFJbqdo= )
```

Snow666


```
                SIG    A 3 3 3600 20001008023531 (20000908023531 12345
                       meudominio.com.BDwfBm2j6xFLoXttzvtuln9ZD+9q
                       UWBAwSBJVBO6WJ/Rc6+Flubj/fs= )

$TTL 7200 ; 2 horas
                SIG    NXT 3 3 7200 20001008023531 (20000908023531 12345
                       meudominio.com.BIMwxryI8NyfWupBe4JJmeRCCj1/
                       FnyPjxAuBOQKTRXX4FsaDrma1X4= )
                NXT    awesome ( A MX SIG NXT )
$TTL 3600 ; 1 hora
                A      128.138.242.1
                A      128.138.243.140
                MX     10 anchor
                MX     99 meudominio.com.
...
```

As zonas assinadas são horríveis. Normalmente elas são quatro vezes maiores do que a zona original e aquele belo aspecto de ordenação lógica é perdido. Basicamente, um arquivo de zonas assinado não pode ser lido por nós seres humanos nem editado manualmente devido aos registros SIG e NXT. Nenhuma parte interna útil para o usuário!

Um registro SIG contém uma grande quantidade de informações:

- O tipo de conjunto de registros que está sendo assinado
- O algoritmo de assinatura que está sendo usado (em nosso caso, ele é 3, o algoritmo DSA)
- O TTL do conjunto de registros que foi assinado
- O momento em que a assinatura expira (no formato *aaaammddhhssss*)
- O momento em que o conjunto de registros foi assinado (também no formato *aaaammddhhssss*)
- O identificador de chaves (em nosso caso, 12345)
- O nome do assinante (meudominio.com.)
- E, finalmente, a assinatura digital em si

Para usar a zona assinada, modifique o parâmetro `file` na instrução `zone` de **named.conf** para meudominio.com para apontar para **db.meudominio.signed** em vez de **db.meudominio**. No BIND 8, você também tem de incluir uma instrução `pubkey` na instrução `zone`; o BIND 8 verifica os dados da zona enquanto ele carrega e, portanto, tem de saber a chave de antemão. O BIND 9 não executa essa verificação. Ele obtém a chave pública do registro KEY nos dados de zona e não precisa de qualquer outra configuração. Uau! É isso aí.

As assinaturas digitais são ótimas para respostas positivas como "Eis o endereço IP para o host anchor.cs.colorado.edu, juntamente com uma assinatura para provar que ele realmente proveio de cs.colorado.edu e de que os dados são válidos". Mas o que acontece com respostas negativas como "Não encontrado nenhum host com estas especificações?". Tais respostas negativas normalmente não retornam nenhum registro assinável.

No DNSSEC, esse problema é tratado por registros NXT que listam o próximo registro na zona em uma ordem canônica.[22] Se o próximo registro após anchor em cs.colorado.edu fosse awesome.cs.colorado.edu e uma consulta por anthill.cs.colorado.edu chegasse, a resposta seria um registro NXT assinado como

```
anchor.cs.colorado.edu. IN NXT awesome.cs.colorado.edu A MX NXT
```

Este registro diz que o nome imediatamente após anchor na zona cs.colorado.edu é awesome e que anchor possui pelo menos um registro A, um registro MX e um registro NXT. O último registro NXT numa zona retorna para o primeiro nome da zona. Por exemplo, o registro NXT para zamboni.cs.colorado.edu apontaria de volta para o primeiro registro, aquele do próprio cs.colorado.edu:

```
zamboni.cs.colorado.edu. IN NXT cs.colorado.edu A MX NXT
```

Registros NXT também são retornados caso o host exista, mas o tipo de registro consultado não existe. Por exemplo, se a consulta fosse para buscar um registro LOC para anchor, o mesmo registro NXT de anchor seria retornado e mostraria somente registros A, MX e NXT.

22. A ordem de certo modo é alfabética, porém com nomes que se encontram mais acima na árvore DNS vindo em primeiro lugar. Por exemplo, na zona cs.colorado.edu, cs.colorado.edu vem antes de qualquer host cs.colorado.edu. Em um nível hierárquico, a ordem é alfabética.

Snow666

Registros NXT fornecem uma maneira fácil de percorrer a zona, dando a um consultador persistente uma transferência de zonas completa, um registro por vez. Nada bom em termos de segurança/privacidade.

O material contido nesta seção descreve o DNSSEC de acordo com o BIND v9.2 (novembro, 2001). A julgar pelas modificações significativas que ocorreram desde a primeira versão do BIND 9, estas informações podem não mais estar corretas quando você estiver lendo este livro. Como sempre, consulte seus manuais e a documentação que acompanha o BIND para os detalhes exatos. Isso posto, vamos dar uma olhada em alguns problemas potenciais com o projeto atual do DNSSEC.

O DNSSEC está em desacordo com as noções de caching e de encaminhadores. O DNSSEC supõe que consultas contactam a zona-raiz primeiro e depois seguem delegações pela cadeia de domínios abaixo até encontrar uma resposta. Cada zona assinada assina as chaves de seus filhos, e a cadeia de confiabilidade é inquebrável e autêntica. Entretanto, quando se usa um encaminhador, a consulta inicial é desviada da zona-raiz e enviada para o seu servidor de encaminhamento para processamento. Um servidor de caching que está consultando por meio de um encaminhador irá verificar novamente as assinaturas, para que as respostas sejam garantidamente seguras. Porém, para que a consulta seja bem-sucedida, o encaminhador tem de ser capaz de retornar todos os SIGs e KEYs necessários para a verificação da assinatura. Servidores não-DNSSEC não sabem como fazer isso, e as RFCs ignoram toda a problemática de encaminhamento.

O BIND 9 implementa alguns recursos extras além daqueles exigidos pela RFC2535 de modo que um servidor de caching no BIND 9 pode usar DNSSEC por meio de um encaminhador BIND 9. Se estiver usando encaminhadores e quiser utilizar DNSSEC, talvez você tenha de empregar o BIND 9 em seu site.

Infelizmente, esses sites carregados que usam encaminhadores e caching são provavelmente os mais interessados em DNSSEC. Infelizmente, os criadores de padrões não pensam muito em todas as implicações para as demais partes do sistema DNS.

O DNSSEC também depende da existência de uma infra-estrutura de chaves públicas que ainda não é bem uma realidade. Não há nenhuma forma tranqüila para fazer com que o pai assine as chaves de um filho; não podemos enviar um e-mail para hostmaster@com e obter chaves assinadas de volta. Nos próximos anos devemos começar a ver o DNSSEC sendo empregado, provavelmente começando com versões assinadas da zona-raiz e as zonas-primeiro nível de código de países na Europa. Os administradores de sistemas precisam ficar de olho nos avanços do DNSSEC, porém, quando estávamos escrevendo este livro, ainda era muito cedo para realmente nos preocuparmos com ele.

Uma outra questão problemática em relação ao DNSSEC é que ele aumenta o tamanho dos pacotes de resposta, pois, para cada resposta de dados, também precisamos incluir registros SIG e NXT. O limite de UDP de 512 bytes começará a vir muito mais cedo. À medida que o EDNS0 for empregado e um número maior de servidores de nomes for capaz de negociar tamanhos de pacotes maiores, esta questão deve, progressivamente, se tornar menos ameaçadora.

As assinaturas de transações (TSIG/TKEY) usam menos tempo de CPU e largura de banda do que a autenticação de chaves públicas, mas elas garantem somente que você saiba de onde provêm suas respostas, e não que estejam corretas. Uma combinação de um relacionamento TSIG com um servidor que se sabe que atende completamente ao DNSSEC poderia fornecer um grau razoável de segurança. Não é possível ter um relacionamento TSIG com cada servidor que você possa vir a querer conversar, já que os relacionamentos TSIG têm de ser configurados manualmente.

Microsoft ruim, Linux bom

O Windows 2000 usa registros de recursos SRV para descobrir tudo: servidores de nomes, impressoras, sistemas de arquivos e assim por diante. Eles seguem as especificações IETF em suas implementações de registros SRV, porém a maneira pela qual inserem os registros no DNS usando uma atualização dinâmica segura não é padrão. A Microsoft utiliza uma variação de assinaturas de transação chamada GSS-TSIG que também se baseia num segredo compartilhado. O segredo compartilhado é obtido pelo Kerberos a partir do Kerberos KDC (Key Distribution Center). No momento, a implementação da Microsoft usa o campo de extensões do fornecedor e, conseqüentemente, não é compatível com a versão de fonte aberto do Kerberos 5. (Huumm... Envolver, estender, exterminar.)

Se quiser rodar Win2K e usar registros SRV, você terá de destruir o seu domínio Kerberos existente e rodar um servidor Kerberos Win2K em suas redes. Para alguns sites com uma infra-estrutura Kerberos rica, esse problema é uma interrupção daquilo que está dando certo. A Microsoft parece estar usando protocolos abertos o suficiente apenas para surrupiar (atendendo furtivamente) listas de verificação de compras de empresas, mas não o suficiente para permitir que alguém interopere e venda em seu mercado. Espera-se que a Microsoft documente suas extensões sem o acordo de não-revelação exigido atualmente para que o ISC possa fazer com que seu TSIG e o GSS-TSIG da Microsoft interoperem.

Snow666

Uma semana depois de o Win2K ter sido lançado, a carga de consultas nos servidores-raiz DNS aumentou significativamente. Um pouco de investigação revelou que computadores com Win2K mal configurado estavam tentando atualizar dinamicamente as zonas-raiz ou de primeiro nível. O número de consultas UDP ao servidor-raiz A atingiu mais do que o dobro como conseqüência disso. Para piorar as coisas ainda mais, quando suas solicitações de atualização eram recusadas, os computadores com Win2K solicitavam um registro KEY para tentar uma atualização autenticada. Isso também falhou, de modo que eles tentaram uma última vez abrindo uma conexão TCP para fazer uma atualização dinâmica autenticada. Um servidor-raiz não tem tempo para atender às zilhões de solicitações de conexões TCP resultantes. Quando estávamos para editar o livro, a situação era um pouco melhor, mas ainda não totalmente solucionada, com operadores de servidores raiz apontando para a Microsoft e dizendo "Não, não, nós não!".

Para fazer com que os computadores com Win2K parem de tentar atualizar as zonas-raiz e se comportem melhor com seus servidores de nomes Linux, proceda da seguinte maneira:

- Dê um clique com o botão direito do mouse em Meus Locais de Redes e selecione Propriedades, que exibirá uma janela chamada Rede e Conexões Discadas.
- Dê um clique com o botão direito do mouse em cada conexão alternadamente e selecione Propriedades.
- Clique sobre Internet Protocol (TCP/IP), depois clique sobre o botão Propriedades.
- Clique no botão Configurações Avançadas... na parte inferior da página de propriedades.
- Clique sobre a guia DNS na parte superior.
- Na parte inferior da página, desmarque a caixa "Registre o endereço desta conexão em DNS".
- Vá clicando em OK até chegar ao fim.

Talvez você também descubra seus sistemas Win2K inundando a rede com consultas DNS. O Win2K não segue a especificação e consulta somente uma fonte por vez, mas, ao contrário, envia consultas paralelas fazendo a mesma pergunta, com timeouts muito baixos. Como padrão, cada computador Win2K é configurado para ser um servidor de nomes de caching com hábitos sórdidos. Você pode desativar esse comportamento com a seguinte seqüência de passos:

- Selecione Configurações->Painel de controle, no menu Iniciar.
- No Painel de controle, dê um clique duplo em Ferramentas Administrativas.
- Na página Ferramentas Administrativas, dê um clique duplo em Gerenciamento de Computador.
- Do lado esquerdo, clique no sinal "+" próximo a Serviços e Aplicativos para expandi-lo.
- Selecione Serviços no lado esquerdo e dê um clique duplo em Cliente DNS no lado direito.
- Clique no menu suspenso ao lado de "Tipo de inicialização" e selecione Manual.
- Clique no botão Parar para interromper o cliente DNS já em execução.
- Clique em OK até o final.

Atualmente, é melhor evitar a implementação DNS da Microsoft o máximo possível. Provavelmente você vai querer apontar para seus computadores com Win2K num servidor de nomes Linux para ganhar total funcionalidade DNS sem as rajadas mortais de tráfego desnecessário geradas pela "criatividade" da Microsoft.

Os servidores Microsoft geram um tumulto quando consultam uma zona cujos servidores estão fora de combate. Se seu site faz delegações a servidores que parecem estar inativos, é uma boa pratica pelo menos instalar um arquivo de zonas vazio que substitua servidores faltantes. Essa medida impedirá que servidores Microsoft que consultam o servidor defeituoso acabem ficando confusos e "surrando" os servidores-raiz e gTLD.

16.14 Testando e Depurando

named fornece várias ferramentas de depuração já embutidas, destacando-se entre elas o sistema de registro (log) bastante configurável. Você pode especificar níveis de depuração na linha de comando ou configurá-los com **ndc/rndc**. Você também pode instruir **named** para fazer um dump das estatísticas de operação para um arquivo e verificar pesquisas de nomes com ferramentas externas como o **dig**.

Veja o Capítulo 11 para obter maiores informações sobre syslog.

Registro

Os recursos de registro de **named** são suficientemente flexíveis para deixar você de cabelo em pé. O BIND 4 usava syslog para registrar mensagens de erro e anomalias. O BIND 8 generaliza os conceitos de syslog adicionando uma outra camada indireta e de suporte

Snow666

para registro diretamente em arquivos. Antes de mergulharmos no assunto, vamos dar uma olhada no miniglossário de termos de registro do BIND mostrado na Tabela 16.13.

Tabela 16.13 Um léxico de registro do BIND.

Termo	Significado
canal	Local onde mensagens podem ser enviadas: syslog, um arquivo ou **/dev/null**.
categoria	Uma classe de mensagens que **named** é capaz de gerar, por exemplo, mensagens sobre atualizações dinâmicas ou mensagens sobre respostas a consultas.
módulo	O nome do módulo de origem que gera uma mensagem (BIND 9 somente).
facilidade	Um nome de recurso syslog. O DNS não possui um recurso próprio específico, porém você pode escolher entre as várias opções-padrão.
gravidade	A gravidade de uma mensagem de erro, o que o syslog chama de prioridade.

Configuramos o registro do BIND com uma instrução `logging` em **named.conf**. Primeiramente definimos os canais, os possíveis destinos para as mensagens. A seguir informamos as várias categorias de mensagem que vão para determinados canais.

Quando uma mensagem é gerada, lhe são atribuídos uma categoria, um módulo (no BIND 9) e uma gravidade em seu ponto de origem. Ela é então distribuída para todos os canais associados à sua categoria e módulo. Cada canal possui um filtro de gravidade que indica que nível de gravidade uma mensagem deve ter para poder passar. Canais que levam o syslog tambem são filtrados de acordo com as regras contidas em **/etc/syslog.conf**.

Eis a descrição geral de uma instrução `logging`:

```
logging {
   definição_do_canal;
   definição_do_canal;
   …
   category nome_da_categoria {
      nome_do_canal;
      nome_do_canal;
      …
   };
};
```

Uma *definição_do_canal* tem um aspecto ligeiramente diverso dependendo de o canal ser um canal de arquivos ou um canal de syslog. Você tem de escolher `file` ou `syslog` para cada canal; um canal não pode se comportar de ambas as maneiras simultaneamente.

```
channel nome_do_canal {

   file caminho [versions numvers | unlimited] [size especif-tamanho];
   syslog recurso;

   severity gravidade;
   print-category yes | no;
   print-severity yes | no;
   print-time yes | no;
};
```

Consulte a página 158 para uma lista dos nomes de recursos syslog.

Para um canal de arquivos, *numvers* informa o números de versões de backup de um arquivo que devem ser mantidas e *especif-tamanho* especifica qual o tamanho máximo permitido para o arquivo (exemplos: `2048`, `100k`, `20m`, `15g`, `unlimited`, `default`) antes dele automaticamente sofrer uma rotação. Se você chamar um canal de arquivos de **mylog**, as versões rotacionadas serão **mylog.0**, **mylog.1** e assim por diante.

No caso de syslog, *recurso* especifica qual o nome do recurso usado para registrar em log a mensagem. Poderá ser qualquer recurso-padrão. Na prática, somente `daemon` e `local0` a `local7` são opções razoáveis.

O restante das instruções numa *definição_do_canal* é opcional. *gravidade* também pode ter os valores (em ordem descendente) `critical`, `error`, `warning`, `notice`, `info` ou `debug` (com um nível numérico opcional, por exemplo, `severity debug 3`). O valor `dynamic` também é reconhecido e concorda com o nível de depuração atual do servidor.

Snow666

As diversas opções `print` acrescentam ou suprimem prefixos de mensagens. Syslog anexa o horário e o host relator a cada mensagem registrada em log, mas não coloca informações relativas à gravidade ou à categoria. No BIND 9, o nome de arquivo de origem (módulo) que gerou a mensagem também está disponível na forma de uma opção `print`. Faz sentido habilitar `print-time` somente para canais de arquivos; não há nenhuma necessidade de duplicar o horário do syslog.

Os quatro canais listados na Tabela 16.14 são predefinidos por padrão. Os canais-padrão devem ser suficientes para a maioria das instalações.

Tabela 16.14 Canais de registro predefinidos no BIND.

Nome do canal	O que ele faz
default_syslog	Envia ao syslog, daemon de recursos, informações de gravidade
default_debug	Registra informações de log no arquivo **named.run**, gravidade configurada em dinâmica
default_stderr	Envia ao erro-padrão do processo **named** informações de gravidade
null	Descarta todas as mensagens

A Tabela 16.15 mostra a lista atual de categorias de mensagens para o BIND 8 e 9. As categorias do BIND 9 estão numa certa flutuação; refira-se aos arquivos de código-fonte

```
lib/dns/include/dns/log.h
bin/named/include/named/log.h
```

para ter uma lista das categorias atuais.

Tabela 16.15 Categorias de registro no BIND.

Categoria	Versão	O que ela inclui
`cname`	8	Mensagens na forma "... aponta para um CNAME"
`eventlib`	8	Depuração de informações do sistema de eventos[a]
`insist`	8	Falhas de verificação na consistência interna
`load`	8	Mensagens de carregamento de zonas
`maintenance`	8	Eventos de manutenção periódicos
`ncache`	8	Mensagens sobre caching negativo
`os`	8	Problemas de sistema operacional
`packet`	8	Dumps de pacotes recebidos e enviados[a]
`panic`	8	Erros fatais (duplicados nesta categoria)
`parser`	8	Processamento de arquivo de configuração de baixo nível
`response-checks`	8	Comentário sobre pacotes de resposta malformados ou inválidos
`config`	8/9	Processamento e análise sintática de arquivo de configuração
`db/database`	8/9	Mensagens sobre operações em bancos de dados
`default`	8/9	Categorias sem nenhuma atribuição de canal explícita[b]
`lame-servers`	8/9	Servidores que supostamente devem servir uma zona, mas não o fazem[c]
`notify`	8/9	Mensagens sobre o protocolo de notificação de "zona modificada"
`queries`	8/9	Uma mensagem de log curta para cada consulta que o servidor recebe (!)
`security`	8/9	Solicitações aprovadas/desaprovadas
`statistics`	8/9	Estatísticas agregadas de servidor de nomes
`update`	8/9	Mensagens sobre atualizações dinâmicas
`xfer-in`	8/9	Transferências de zona que o servidor está recebendo
`xfer-out`	8/9	Transferências de zona que o servidor está enviando
`client`	9	Solicitações de clientes
`dnssec`	9	Mensagens DNSSEC

Snow666

Tabela 16.15 Categorias de registro no BIND. (*continuação*)

Categoria	Versão	O que ela inclui
`general`	9	Mensagens sem classificação
`network`	9	Operações de rede
`resolver`	9	Resolução DNS, por exemplo, pesquisas recursivas para clientes

a. Tem de ser um único canal de arquivos.
b. A categoria `default` também é a categoria que captura tudo para mensagens sem classificação no BIND 8.
c. Tanto a zona-pai quanto a zona-filha poderia estar com falhas; é impossível afirmar sem investigação.

A configuração de registro-padrão para o BIND 8 é

```
logging {
   category default { default_syslog; default_debug; };
   category panic { default_syslog; default_stderr; };
   category eventlib { default_debug; };
   category packet { default_debug; };
};
```

e para o BIND 9 ele é

```
logging {
   category default { default_syslog; default_debug; };
};
```

Você deve observar os arquivos de log ao fazer modificações no BIND, e talvez aumentar o nível de registro. A seguir, reconfigure para preservar somente mensagens graves assim que **named** estiver estabilizado.

Registro de consultas pode ser bastante educativo. Você pode verificar se suas instruções `allow` estão funcionando, veja quem está lhe consultando, identifique clientes problemáticos etc. Gravar no syslog é menos eficiente do que gravar diretamente num arquivo; portanto, use um canal de arquivos em um disco local quando estiver registrando em log todas as consultas. Tenha reservado uma boa quantidade de espaço em disco e esteja pronto para desativar o registro de consultas assim que já tiver dados suficientes. O registro de consultas é uma boa verificação para ser feita após grandes mudanças na configuração, especialmente se você tiver anotações de como sua carga de consulta costumava ser antes das alterações. Para iniciar o registro de consultas, simplesmente direcione a categoria `queries` para um canal. **rndc querylog** ativa/desativa o registro de consultas no BIND 9.

Algumas mensagens de log comuns são listadas a seguir:

- *Lame server*. Se obtiver esta mensagem sobre uma de suas próprias zonas, você configurou algo incorretamente. A mensagem é relativamente inofensiva se for sobre alguma zona externa, na Internet; trata-se de problema alheio. Trata-se de uma boa mensagem para jogar fora direcionando-a ao canal nulo.
- *Bad referral*. Esta mensagem indica uma comunicação incorreta entre os servidores de nomes de uma zona.
- *Not authoritative for*. Um servidor escravo é incapaz de obter dados autorizados para uma zona. Talvez ele esteja apontando para o mestre incorreto ou talvez o mestre teve problemas para carregar a zona em questão.
- *Rejected zone*. **named** rejeitou um arquivo de zonas, pois continha erros.
- *No NS RRs found*. Um arquivo de zonas não tinha registros NS após o registro SOA. Poderia ser que os registros estivessem faltando ou que não começassem com tabulação ou espaço em branco. No último caso, os registros não são anexados à zona do registro SOA e são, conseqüentemente, mal interpretados.
- *No default TTL set*. A maneira preferida para configurar o TTL padrão é com uma instrução `$TTL` no topo do arquivo de zonas. Essa mensagem de erros indica que o `$TTL` está faltando. No BIND 8 ele usa como padrão o valor do parâmetro *mínimo* com o registro SOA.[23] No BIND 9, o $TTL é exigido; **named** se recusa a carregar arquivos de zonas que não especificam um `$TTL`.
- *No root nameserver for class*. Seu servidor está tendo problemas em encontrar os servidores de nomes raiz. Verifique seu arquivo de dicas e a conectividade do servidor com a Internet.
- *Address already in use*. A porta na qual **named** quer executar já está sendo usada por um outro processo, provavelmente uma outra cópia de **named**. Se você não encontrar nenhum outro **named**, ele pode ter tra-

23. O significado de *mínimo* mudou no BIND 8.2, do TTL padrão para todos os registros para o padrão TTL para caching negativo.

Snow666

vado e deixado um socket de controle **ndc** aberto, o qual você terá de rastrear e remover.[24] Uma boa maneira de corrigir isto é parar e reiniciar o processo **named**:

```
% sudo /etc/init.d/named stop (or /sbin/service named stop)
% sudo /etc/init.d/named start (or /sbin/service named start)
```

- *Dropping source port zero packet from …* Versões recentes do BIND permitem que você configure o número da porta de origem de consultas, e os administrador de sistemas usam este recurso para acrescentar regras a seus firewalls que podem reconhecer seus pacotes DNS pela porta de origem. Entretanto, 0 é um valor ilegal para um número de porta TCP/UDP. Se a mensagem de erro estiver relacionada a um de seus hosts, você deve alterar a diretiva `query-source` no arquivo **named.conf** do host para corrigir esse erro.
- *Denied update from […] for …* Uma atualização dinâmica para uma zona foi tentada e negada devido à cláusula `allow-update` ou `update-policy` em **named.conf** para esta zona.

Você pode encontrar uma tabela excelente de mensagens de erros BIND em

http://www.acmebw.com/askmrdns/bind-messages.html

Níveis de depuração

Os níveis de depuração **named** são indicados pelos inteiros de 0 a 11. Quanto maior o número, mais longa a saída. O nível 0 desativa a depuração. Os níveis 1 e 2 são bons para depurar sua configuração e banco de dados. Níveis acima próximos de 4 são apropriados para aqueles que fazem a manutenção do código.

A depuração é acionada na linha de comando **named** com o flag **-d**. Por exemplo,

```
# named -d2
```

iniciaria **named** no nível de depuração 2. Como padrão, as informações de depuração são gravadas no arquivo **named.run** no diretório de trabalho atual por meio do qual **named** é iniciado. O arquivo **named.run** cresce muito rapidamente; portanto, não saia para tomar uma cerveja enquanto estiver no processo de depuração, caso contrário terá problemas maiores ao retornar.

Você também pode ativar a depuração enquanto **named** estiver em execução através de **ndc trace**, que incrementa o nível de depuração em 1. **ndc notrace** desativa completamente a depuração. (Infelizmente, **rndc** não suporta esses comandos.) Você também pode ativar a depuração definindo um canal de registro que inclua uma especificação de gravidade como

```
severity debug 3
```

que enviará todas as mensagens de depuração até o nível 3 para um determinado canal. Outras linhas na definição do canal especificam o destino dessas mensagens de depuração. Quanto maior o nível de gravidade, mais informações são registradas em log.

Observar os logs ou a saída de depuração ilustra com que freqüência os dados DNS são mal configurados. Aquele maldito pontinho no final dos nomes (ou melhor, a falta dele) é responsável por um nível alarmante de tráfego DNS. Teoricamente, o ponto é necessário no final de cada nome de domínio totalmente qualificado.

Depurando com o ndc

O comando **ndc** (que se chama **rndc** no BIND 9) é uma ferramenta útil para manipular **named**. A Tabela 16.16 ilustra algumas das opções que ele aceita. Comandos que geram arquivos os colocam no diretório especificado como diretório inicial de **named** em **named.conf**.

Tabela 16.16 Comandos ndc e rndc úteis.

Comando	Função
Help	Lista os comandos **ndc** disponíveis
Status	Exibe o estado atual do processo **named** em execução
start (V8)	Inicia **named**
Stop	Interrompe **named** e na V9 salva as atualizações pendentes
halt (V9)	Interrompe **named** sem realizar atualizações pendentes
trace	Incrementa o nível de depuração em uma unidade[a]
notrace	Desativa a depuração[a]

24. Num servidor gTLD, essa mensagem provavelmente significa que com ainda está carregando. :-)

Snow666

Tabela 16.16 Comandos ndc e rndc úteis. (*continuação*)

Comando	Função
dumpdb	Faz um dump do banco de dados DNS para **named_dump.db**
stats	Faz um dump das estatísticas para **named.stats**
flush (V9)	Esvazia o cache do servidor
reload	Recarrega **named.conf** e arquivos de zonas
reload *zona*	Recarrega somente a *zona* especificada
reconfig	Recarrega o arquivo de configuração e quaisquer zonas novas
querylog	Ativa/desativa rastreamento de consultas que chegam

a. Isso não acontece no **rndc**.

ndc reload faz com que **named** releia seu arquivo de configuração e recarregue arquivos de zonas. O comando **ndc reload** *zona* é útil, especialmente num servidor muito ativo, quando somente uma zona foi modificada e você não deseja recarregar todas as zonas.

ndc dumpdb faz com que **named** gere um dump de seu banco de dados para **named_dump.db**. O arquivo de dump é grande e inclui não somente dados locais, como também quaisquer dados colocados em cache que o servidor de nomes tenha acumulado. Um dump recente do cache do banco de dados em nosso servidor principal colorado.edu ocupava mais de 16MB, porém os dados de zona carregados ocupavam menos do que 200K. Muito caching nesse caso.

Em situações de DNS dividido em que temos um processo **named** manipulando a zona interna e um outro manipulando a zona externa, o **ndc** não lhe dá controle suficiente. Você não pode especificar o PID do processo **named** que deseja controlar. Da mesma forma, se estiver executando **named** num ambiente em que foi aplicado **chroot**, você terá de dar a **ndc** o caminho para o socket **ndc**, pois ele não estará em **/var/run** que é o local onde **ndc** esperaria que ele estivesse. Use algo como:

```
# sudo ndc -l /var/named/var/run/ndc
```

No BIND 9, é importante ter certeza de que a versão de **named** e a versão de **rndc** sejam coincidentes, a fim de não receber uma mensagem de erro sobre uma discordância entre versão de protocolos. Simplesmente instale ambos ao fazer uma atualização. Cada nova versão faz com que **rndc** fique mais próximo da funcionalidade oferecida por **ndc** no BIND 8. Sinais no BIND 9 não funcionam mais para controlar **named**. SIGHUP irá recarregar o arquivo de configuração e os arquivos de zona; qualquer outro sinal destruirá o processo **named** em execução.

Estatísticas no BIND 8

A versão BIND 8 de **named** mantém um registro de estatísticas de consultas que você poderá acessar por meio de **ndc stats**. Executar esse comando faz com que **named** grave as informações estatísticas no arquivo **named.stats**. Um exemplo de arquivo estatístico de um servidor cs.colorado.edu que esteve ativo por 83 dias é mostrado a seguir. Essas informações são normalmente impressas numa coluna longa, porém nós a compactamos um pouco eliminando entradas de tipos de registros de recursos obsoletos ou não utilizados. Também reformatamos a seção inferior; normalmente ela é uma longa linha com valores.

```
+++ Statistics Dump +++ (1008202999) Wed Dec 12 17:23:19 2001

2612979      time since boot (secs)
2612979      time since reset (secs)
9            Unknown query types
1230669      A queries
7            NS queries
0            CNAME queries
8075         SOA queries
705639       PTR queries
41366        MX queries
384370       TXT queries
7198         AAAA queries
6            SRV queries
353057       ANY queries

++ Name Server Statistics ++
```

Snow666


```
RR       RNXD   RFwdR   RDupR  Rfail  RFErr  Rerr   RAXFR  RLame
428444   33365  163125  21179  112775 0      450    0      27388

Ropts    SSysQ  Sans    SFwdQ  SDupQ  Serr   RQ      RIQ    RFwdQ
0        84483  2587274 273517 520533 0      2730553 0      273517

RDupQ    RTCP   SFwdR   Sfail  SFErr  SNaAns SNXD
3272     26671  163125  0      0      753041 166469
```

O número de consultas TXT parece estar anomalamente alto. Inicialmente pensamos que alguém estivesse sondando endereços para spam ou máquinas para invadir. Mas depois nos lembramos de que o campus usa a classe Hesiod para armazenar informações de diretório de e-mail e nome de e-mail (*primeiro.último*) para cada usuário nos registros TXT.

Os dados confusos no final contabilizam coisas como o número de consultas duplicadas, respostas duplicadas e delegações incorretas vistas por esse servidor. A letra inicial nas abreviaturas significa recebida (`R, de received`) ou enviada (`S, de sent`); a letra final indica consulta (`Q, de query`) ou resposta (`R, de resposta`). O real significado desses cabeçalhos está documentado no código, no qual um breve comentário identifica cada abreviatura — veja o arquivo **ns_stats.c** sob o diretório **bin/named** na distribuição do BIND 8. Uma maneira menos dolorosa de decodificar essas abreviaturas é ler a seção sobre estatísticas no livro DNS da O'Reilly.

Qualquer consulta que resultar num erro é colocada em log, contabilizada em um ou mais coletores estatísticos e perdida. A categoria `Unknown query types` inclui qualquer consulta para um tipo de registro de recursos que o servidor não reconhece. À medida que novos tipos de registros são padronizados, eles aparecem nessa coluna até que os servidores de nomes sejam atualizados para incluir os novos tipos de registros. O coletor `ANY` não é realmente um tipo de registro de recursos; ele contabiliza consultas que indaguem sobre toda e qualquer informação que um servidor possa ter sobre um determinado nome. As entradas `Dup` na metade inferior do relatório representam consultas ou respostas duplicadas. As duplicações normalmente ocorrem quando uma consulta é expirada antes de sua resposta ter sido recebida; o consultante então submete novamente a consulta.

No BIND 8, **ndc stats** também gera informações estatísticas de memória no arquivo **named.memstats** caso `deallocate-on-exit` esteja ativo. As informações estatísticas de memória no BIND 9 podem ser acessadas somente pelo modo de depuração de **named**.

Estatísticas no BIND 9

O BIND 9 não coleta informações estatísticas de consultas na mesma proporção que o BIND 8, porém ele mantém algumas informações sintetizadas. **named** faz um dump das informações para o arquivo **named.stats** em seu diretório de trabalho ao receber uma sinalização de **rndc**:

```
% sudo rndc stats
```

Veja um exemplo da saída gerada:

```
+++ Statistics Dump +++ (1008196810)
success 30647
referral 18
nxrrset 1405
nxdomain 6833
recursion 13630
failure 1699
--- Statistics Dump --- (1008196810)
```

As estatísticas mostram o número de sucesso *versus* falha nas pesquisas e categorizam os diversos tipos de erros. Este servidor respondeu a 30.647 pesquisas com sucesso e falhou em responder 9.937 vezes. O maior contribuinte para as falhas foi o erro `nxdomain`, que ocorre quando o nome de host solicitado é desconhecido.

Os 6.833 erros nxdomain seriam de longe em maior número se sua fonte primária fosse erros de digitação e de ortografia cometidos pelos usuários. Neste caso, os erros realmente derivam de um bug no solucionador da Microsoft[25] que envia muitas consultas para registros A (de endereços) com o campo de dados configurado para um endereço IP em vez de um nome de host conforme exigido pela especificação DNS. Estas consultas bizarras parecem ter um domínio de primeiro nível algo como 56 (ou seja lá qual for o último byte da string de endereço IP), de modo que

25. A Microsoft corrigiu esse erro no Windows 2000 Service Pack 2.

Snow666

elas fazem com que o servidor de DNS local incomode os servidores raiz sobre este domínio inexistente, levando, em última instância, a um erro nxdomain.[26]

A entrada `failure` contabiliza as falhas que não são nem `nxdomain` (no such domain, não existe tal domínio) nem `nxrrset` (no such resource record set, não existe tal conjunto de registros de recursos).

Depurando com nslookup, dig e host

Três ferramentas podem ser usadas por meio do shell para consultar o banco de dados DNS: **nslookup**, **dig**, e **host**. Cada uma delas é distribuída com o BIND 8. **nslookup** é a mais antiga destas ferramentas e sempre fez parte da distribuição BIND. **dig**, o buscador de informações de domínio, foi originalmente escrito por Steve Hotz; ele foi reescrito para o BIND 9 por Michael Sawyer e também acompanha o BIND 9. **host**, de Eric Wassenaar, é mais uma ferramenta de código-fonte aberto; ela dispõe de uma saída amigável e funções para verificar a sintaxe de seus arquivos de zonas.

Discutimos cada uma destas ferramentas a seguir, porém recomendamos **dig** em relação a **nslookup**; gostamos de **host** também. Pode ser que algumas vezes você obtenha resultados diferentes destas ferramentas devido a bibliotecas de solucionador diferentes que cada uma utiliza. **dig** e **host** usam o solucionador do BIND e **nslookup** possui a sua própria ferramenta.

nslookup é um comando em nível de usuário que consulta o banco de dados DNS. Ele espera receber nomes totalmente qualificados terminando num ponto e anexa o domínio-padrão caso você se esqueça do ponto. Para nomes locais, este comportamento normalmente é o desejado. A Tabela 16.17 apresenta uma lista curta de comandos **nslookup**.

Tabela 16.17 Comandos compreendidos por nslookup.

Comando	Função
nome	Imprime informações sobre o *nome* do host ou do domínio
help ou **?**	Mostra uma lista completa de comandos
exit	Abandona
server *host*	Configura o servidor-padrão, usando o servidor atual
lserver *host*	Configura o servidor-padrão, usando o servidor inicial
set type= xxx	Configura os tipos de registros a serem consultados[a]
set debug	Ativa a depuração
set d2	Ativa um grande nível de depuração
ls *domínio*	Lista todas as associações host/endereço

a. **any** é um bom valor que significa "todos os tipos de registros".

dig fornece, basicamente, a mesma funcionalidade que **nslookup**, porém possui padrões mais sensíveis, fornece mais informações e possui uma interface com o usuário melhor (especialmente quando comparada a versões mais antigas do **nslookup**).

Por exemplo, para solicitar registros MX de anchor, use

```
% dig anchor.cs.colorado.edu. mx
```

O comando

```
% dig @ns1.berkeley.edu vangogh.berkeley.edu. any
```

obtém registros completos de vangogh de um servidor de berkeley.edu e

```
% dig -x 128.32.33.5
```

faz uma pesquisa reversa para vangogh. Tome cuidado com a sintaxe *@servidor-de-nomes.algum-lugar*; se o servidor não existir, **dig** usa seu servidor local e retorna a resposta para você. Esse é um excelente comportamento para backup, mas nem sempre você vai querê-lo.

26. A gravidade do problema é até mais evidente quando consideramos que nesta instância, todas as consultas incorretas foram geradas por uma *única* máquina rodando o Windows 2000.

Snow666

Eis um exemplo que usa **host**, **dig** e **nslookup** para obter os mesmos dados:

```
% nslookup
Default Server: bb.rc.vix.com
Address: 204.152.187.11
> set type=any
> amazon.com.
Server: bb.rc.vix.com
Address: 204.152.187.11
Non-authoritative answer:
amazon.com            nameserver = AUTHOO.NS.UU.NET
amazon.com            nameserver = NS2.PNAP.NET
amazon.com            nameserver = NS1.PNAP.NET
amazon.com            nameserver = NS-1.amazon.com
amazon.com            preference = 10, mail exchanger = service-4.amazon.com
amazon.com            preference = 10, mail exchanger = service-5.amazon.com
amazon.com            internet address = 208.216.182.15
Authoritative answers can be found from:
amazon.com            nameserver = AUTHOO.NS.UU.NET
amazon.com            nameserver = NS2.PNAP.NET
amazon.com            nameserver = NS1.PNAP.NET
amazon.com            nameserver = NS-1.amazon.com
AUTHOO.NS.UU.NET      internet address = 198.6.1.65
NS2.PNAP.NET          internet address = 206.253.194.97
NS1.PNAP.NET          internet address = 206.253.194.65
NS-1.amazon.com       internet address = 209.191.164.20
service-4.amazon.com  internet address = 209.191.164.50
service-5.amazon.com  internet address = 209.191.164.51
```

nslookup retorna quatro registros NS, dois registros MX e um registro A. Ele também fornece os endereços IP dos servidores de nomes e hosts MX.

```
% dig amazon.com. any
; <<>> DiG 8.3 <<>> amazon.com any
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 4, ADDITIONAL: 6
;; QUERY SECTION:
;; amazon.com, type = ANY, class = IN
;; ANSWER SECTION:
amazon.com.           1h27m11s     IN NS AUTHOO.NS.UU.NET.
Amazon.com.           1h27m11s     IN NS NS2.PNAP.NET.
Amazon.com.           1h27m11s     IN NS NS1.PNAP.NET.
Amazon.com.           1h27m11s     IN NS NS-1.amazon.com.
Amazon.com.           59m22s       IN MX 10 service-4.amazon.com.
Amazon.com.           59m22s       IN MX 10 service-5.amazon.com.
Amazon.com.           1h59m29s     IN A  208.216.182.15
;; AUTHORITY SECTION:
amazon.com.           1h27m11s     IN NS AUTHOO.NS.UU.NET.
amazon.com.           1h27m11s     IN NS NS2.PNAP.NET.
amazon.com.           1h27m11s     IN NS NS1.PNAP.NET.
amazon.com.           1h27m11s     IN NS NS-1.amazon.com.
;; ADDITIONAL SECTION:
AUTHOO.NS.UU.NET.     13h11m20s    IN A  198.6.1.65
NS2.PNAP.NET.         20h51m44s    IN A  206.253.194.97
NS1.PNAP.NET.         20h51m44s    IN A  206.253.194.65
NS-1.amazon.com.      59m22s       IN A  209.191.164.20
service-4.amazon.com. 59m22s       IN A  209.191.164.50
service-5.amazon.com. 59m22s       IN A  209.191.164.51
;; Total query time: 7 msec
;; FROM: bb.rc.vix.com to SERVER: default -- 204.152.187.11
;; WHEN: Sun Jul 2 12:45:59 2000
;; MSG SIZE sent: 28 rcvd: 338
```

www.snow666.blogspot.com
Snow666

dig gera uma quantidade muito grande de informações. Sua saída inclui não somente as mesmas informações de domínio, como também o número de consultas enviadas e o tempo de ida e volta das respostas. A saída é formatada corretamente para ser usada num arquivo de zonas, que é particularmente prático quando buscamos informações de servidores-raiz para nossos arquivos de dicas.

host gera uma saída concisa-mas-amigável como padrão, porém ele pode ser tornado mais prolixo pela opção **-v** (embora não tão prolixo quanto **dig**). Ele espera que o domínio que você esteja pesquisando termine num ponto. Se você pesquisar um nome relativo, **host** tentará primeiramente anexar os domínios em seu arquivo **resolv.conf**; se nenhum deles funcionar, **host** simplesmente anexará o ponto.

```
% host amazon.com.
amazon.com has address 208.216.182.15
amazon.com mail is handled (pri=10) by service-4.amazon.com
amazon.com mail is handled (pri=10) by service-5.amazon.com
```

Ao testar uma nova configuração, certifique-se de pesquisar dados tanto para hosts locais quanto para hosts remotos. Se você puder acessar um host pelo seu endereço IP, mas não pelo seu nome, provavelmente o DNS é o culpado.

Delegações defeituosas

Ao registrar um nome de domínio, você está solicitando que uma parte da árvore de nomes DNS seja delegada ao seu servidor de nomes primário e ao seu administrador DNS. Se você nunca usa o domínio ou altera os servidores de nomes sem atualizar os registros de adesão do domínio-pai, isso resultará numa "delegação defeituosa".

Os efeitos de uma delegação defeituosa podem ser muito maléficos. Se um usuário tentar consultar um host em seu domínio incorreto, o seu servidor de nomes recusará a consulta. O DNS tentará novamente a consulta centenas de vezes, massacrando tanto o seu servidor mestre como os servidores-raiz. O BIND usa uma "área de penalidade" de servidor defeituoso para ajudar na carga criada pela delegação defeituosa, porém servidores Microsoft não implementam este recurso.

Num arquivo de tamanho equivalente a 3,5MB (em nível de informações) após quase uma semana, cerca de um terço das entradas eram de delegações incorretas. Destas, 16% envolviam consultas aos servidores-raiz, supostamente para domínios inexistentes. Um usuário persistente consultou os servidores-raiz em busca de tokyotopless.net centenas de vezes. Ufa. Eis um exemplo:

```
Jan 29 05:34:52 ipn.caida.org named[223]: Lame server on 'www.games.net' (in
   'GAMES.net'?): [207.82.198.150].53 'NS2.EXODUS.net'
```

Eis como rastreamos o problema com **dig**; truncamos parte da saída prolixa de **dig**:

```
% dig www.games.net.
;; –
;; QUESTIONS:
;; www.games.net,  type = A, class = IN
;; ANSWERS:
www.games.net.     3600  A 209.1.23.92
;; AUTHORITY RECORDS:
games.net.         3600  NS ns.exodus.net.
games.net.         3600  NS ns2.exodus.net.
games.net.         3600  NS ns.pcworld.com.
;; ADDITIONAL RECORDS: …
```

A primeira consulta no servidor local retorna o registro de endereço para www.games.net e uma lista de servidores autorizados.

O servidor em ns.exodus.net funcionava bem quando o consultamos (não exibimos), porém ns2.exodus.net é uma outra história:

```
% dig @ns2.exodus.net www.games.net.
;; QUESTIONS:
;;     www.games.net, type = A, class = IN
;; AUTHORITY RECORDS:
net.               244362 NS F.GTLD-SERVERS.net.
net.               244362 NS J.GTLD-SERVERS.net.
net.               244362 NS K.GTLD-SERVERS.net.
net.               244362 NS A.GTLD-SERVERS.net.
;; –
```

Snow666

ns2 é listado na forma de um servidor autorizado para o domínio, porém ele não retorna nenhum registro e nos refere aos servidores para o domínio de primeiro nível. Conseqüentemente, podemos concluir que ns2.exodus.net está configurado incorretamente. Algumas vezes ao executarmos o comando **dig** em um servidor autorizado numa tentativa de localizar delegações incorretas, **dig** não retorna nenhuma informação; tente executá-lo novamente com o flag **+norecurse** para que possa ver exatamente o que o servidor em questão conhece.

16.15 Finalizações Dispersas

Esta seção inclui algumas finalizações e exemplos dispersos que podiam ter aparecido logo no começo do capítulo, mas para os quais simplesmente não fomos capazes de encontrar um lugar exato. Nós os coletamos aqui, sem nenhuma ordem em particular.

O arquivo de dicas

O arquivo de dicas nutre o cache de **named** com informações sobre os servidores do domínio-raiz. Colocar os servidores-raiz no cache inicializa o processo de pesquisa para todos os demais nomes. Se você não fornecer um arquivo de dicas, o BIND 9 usará uma lista de informações de servidores-raiz embutida em seu código e será capaz de carregar a zona-raiz de qualquer forma desde que os endereços dos servidores-raiz não sejam alterados entre o momento em que o BIND 9 foi instalado e aquele em que você precisará dos servidores de nomes da zona-raiz. Todas as versões anteriores de BIND requerem um arquivo de dicas. (Se você fornecer um arquivo de dicas para o BIND 9, ele prevalecerá em relação às dicas-padrão já embutidas no sistema.)

Os servidores de nomes raiz mudam de tempos em tempos, porém é mais fácil rastreá-los do que costumava ser antigamente, pois todos são nomes de hosts atribuídos no domínio root-servers.net. Use o arquivo de exemplo a seguir somente para referência.

Se você já tem acesso a um sistema com um servidor de nomes ativo, poderá fazer com que **dig** contacte um servidor de nomes raiz e gere o arquivo de dicas para você. O servidor-mestre atualmente é a.root-servers.net, porém qualquer um dos servidores-raiz faria o mesmo:

```
% dig @f.root-servers.net . ns > root.cache
```

Lembre-se do ponto. Se f.root-servers.net não estiver respondendo, você pode executar a consulta sem especificar um servidor em particular:

```
% dig . ns > root.cache
```

A saída será similar; entretanto, você obterá a lista de servidores-raiz do cache de um servidor de nomes local, não de uma fonte autorizada. Isso seria perfeitamente adequado. Mesmo que você não tenha reinicializado o seu servidor de nomes há um ou dois anos, ele vem atualizando seus registros de servidor-raiz periodicamente à medida que seus TTLs expiram. Quando **named** é inicializado, ele recarrega as dicas de um dos servidores-raiz. Portanto, tudo estará bem desde que seu arquivo de dicas contenha ao menos um servidor-raiz válido que possa ser atingido.

Eis o aspecto do arquivo de cache:

```
; <<>> DiG 8.2 <<>> @f.root-servers.net . ns
; Aqui entra um bocado de informações detalhadas geradas por dig na forma de comentários …

.                           1d1h42m IN NS E.ROOT-SERVERS.NET.
.                           1d1h42m IN NS D.ROOT-SERVERS.NET.
.                           1d1h42m IN NS A.ROOT-SERVERS.NET.
.                           1d1h42m IN NS H.ROOT-SERVERS.NET.
…
E.ROOT-SERVERS.NET. 2d1h42m IN A 192.203.230.10
D.ROOT-SERVERS.NET. 2d1h42m IN A 128.8.10.90
A.ROOT-SERVERS.NET. 2d1h42m IN A 198.41.0.4
H.ROOT-SERVERS.NET. 2d1h42m IN A 128.63.2.53
…
```

Note os pontos que iniciam o primeiro conjunto de registros; eles não são pontos voadores aleatórios, mas sim definem o domínio (raiz) para o qual os registros NS se aplicam. Algumas versões de **dig** mostram o TTL em segundos em vez de dias, minutos e segundos.

Snow666

Um arquivo de dicas atual também pode ser obtido via FTP anônimo de rs.internic.net no arquivo domain/named.root. Comentários nessa versão do arquivo de dicas raiz mostram os nomes antigos de servidores-raiz, que aludem sutilmente às suas localizações históricas. Este arquivo também é espelhado em ftp://ftp.nic.mil/domain/named.root[27].

Configuração de localhost

A associação direta para os nomes localhost ou localhost.*dominio* é feita no arquivo de zonas diretas para o domínio. Entretanto, cada servidor normalmente é o mestre para seu próprio domínio localhost reverso. Eis um exemplo de arquivo de zona:

```
@               IN    SOA   cs.colorado.edu. hostmaster.cs.colorado.edu. (
                                1996110801   ; Número Serial
                                3600         ; Atualiza
                                900          ; Tenta novamente
                                3600000      ; Expira
                                14400 )      ; Mínimo
                IN    NS    cs.colorado.edu.
1               IN    PTR   localhost.cs.colorado.edu.
```

A associação reversa para o endereço localhost (127.0.0.1) jamais muda, de modo que os timeouts podem ser grandes. Observe o número de série, que codifica a data; o arquivo foi alterado pela última vez em 1996. Note também que somente o servidor de nomes mestre é listado para o domínio localhost. O significado de `@` aqui é "0.0.127.in-addr.arpa.".

Ferramentas para gerenciamento de hosts

Os arquivos de banco de dados DNS normalmente ultrapassam os domínios político-administrativos locais. Em muitos casos, um controle centralizado rigoroso é inviável. Essa situação apresenta um problema administrativo comum: Como administrar arquivos de dados críticos (porém frágeis) que muitas pessoas não treinadas podem precisar editar em momentos aleatórios? Também seria bom se o departamento de Física não pudesse alterar os registros do departamento de Engenharia e vice-versa.

Se um domínio político contém vários hosts e possui uma equipe administrativa, então um subdomínio é uma boa maneira de distribuir controle. Mas para um pequeno departamento com apenas alguns poucos hosts, um subdomínio não seria necessário.

***addhost** é disponível em ftp.xor.com.*

Esses dados são excelentes candidatos para um banco de dados LDAP que armazena dados de host que abrangem toda a organização e para ferramentas que constroem arquivos de zonas departamentais individuais. Usamos uma ferramenta antiga e caseira chamada **addhost**, mas pretendemos mudar para LDAP tão logo possamos designá-lo como um projeto para um estudante.

Veja a página 504 para obter maiores informações sobre firewalls.

DNS para sistemas que não estão na Internet

Se você não fizer parte da Internet, mas quiser usar o DNS, poderá declarar seu servidor de nomes primário como autorizado para o domínio-raiz. Essa configuração poderia ser apropriada tanto para uma pequena empresa que ainda não está conectada à Internet quanto para uma grande organização que oculta sua estrutura local por trás de um firewall.

Nesta configuração, seu arquivo de dicas deve apontar para servidores de nomes locais, não para os servidores raiz da Internet. Você ainda deve obter um nome de domínio registrado e legitimar endereços IP ou quem sabe usar os endereços privados RFC1918 descritos na página 192.

16.16 Informações Específicas sobre Distribuições

Cada uma de nossas distribuições Linux de referência vem com o BIND como software-padrão para servidor de nomes. Esta seção descreve os detalhes do software de cada distribuição e suas configurações-padrão.

Versões

A Tabela 16.4 sintetiza as versões do BIND que cada distribuição inclui. As distribuições ISC atuais (8.2.5 e 9.2.0) compilam sem problemas no Linux se você estiver propenso a fazer uma atualização. Se estiver usando o BIND 9,

27. O arquivo named.root também pode ser obtido de ftp://ftp.internic.net (N.R.T.)

Snow666

atualizar para o release mais recente e estável é uma boa idéia; novos recursos são adicionados regularmente, bugs são corrigidos e o desempenho otimizado. Entretanto, o BIND 9 ainda é mais lento do que o BIND 8 ou 4.

Arquivos BIND

Um computador Linux que é um cliente BIND especifica seu domínio-padrão, os demais domínios para buscar quando os nomes não são totalmente qualificados e os endereços IP de seus servidores de nomes locais no arquivo **/etc/resolv.conf**. Um host Linux que atua como um servidor BIND também usa os arquivos **named** listados na Tabela 16.18.

Tabela 16.18 Arquivos BIND no Linux.

Arquivo	Diretório	Descrição
resolv.conf	**/etc**	Arquivo de configuração de biblioteca de solucionadores
named	**/usr/sbin**	Daemon de servidor de nomes
named-xfer	**/usr/sbin**	Código de transferência de zonas
named.conf	**/etc**	Arquivo de configuração para o servidor de nomes no Red Hat e SuSE
named.conf	**/etc/bind**	Arquivo de configuração para o servidor de nomes no Debian Linux
named.pid	**/var/run**	ID de processos do processo **named** em andamento
named.run	*diretório*[a]	Saída gerada no modo de depuração
named.stats	*diretório*[a]	Saída de informações estatísticas
named.memstats	*diretório*[a]	Estatísticas sobre uso de memória (somente BIND 8)
named_dump.db	*diretório*[a]	Dump do banco de dados inteiro

a. O diretório especificado em **/etc/named.conf** como diretório inicial para os arquivos do BIND.

A página de manual para **named** informa que os arquivos de depuração e estatísticas estão localizados em **/var/tmp**. Isso é uma mentira, provavelmente um remanescente da página de manual distribuída com o pacote BIND do ISC.

O arquivo de comutação de servidor de nomes

O Linux inclui um arquivo de comutação, **/etc/nsswitch.conf,** que especifica como as associações nome de host-endereços IP devem ser feitas e se o DNS deve ser experimentado em primeiro lugar, em último ou não deve ser usado. Se não houver nenhum arquivo de comutação presente, o comportamento-padrão é

```
hosts: dns [!UNAVAIL=return] files
```

A cláusula `!UNAVAIL` significa que se o DNS estiver disponível porém um nome não for encontrado lá, a tentativa de pesquisa deve falhar em vez de tentar continuar para a próxima entrada (neste caso, o arquivo **/etc/hosts**). Se nenhum servidor de nomes estivesse ativo (como poderia ser o caso durante o processo de boot), o processo de pesquisa *consultaria* o arquivo **hosts**.

As distribuições usadas como exemplo neste livro utilizam vários padrões para pesquisas de host:

```
hosts: db files nisplus nis dns    # Red Hat
hosts: files dns                   # SuSE
hosts: files dns                   # Debian
```

Realmente não há nenhuma "melhor" maneira de configurar as pesquisas. Isso depende de como o seu site é administrado. Em geral, preferimos manter o máximo de informações possível sobre hosts no DNS, e não em arquivos planos ou NIS. Portanto, gostamos do padrão quando não há nenhum arquivo de comutação (`dns` se **named** estiver em execução; caso contrário, `files`). Os padrões para o SuSE e o Debian também são OK, desde que o arquivo **hosts** não seja muito grande.

Arquivos de configuração

O Red Hat e o SuSE fornecem exemplos de arquivos **/etc/named.conf** muito bem comentados. Eles parecem reafirmar os padrões do BIND em muitos casos, porém eles também agregam comentários com o significado das várias opções e onde seus ajustes de configuração diferem dos padrões. Tanto o Red Hat quanto o SuSE usam o diretório de origem, **/var/named**, para abrigar **named**. O Red Hat inclui a documenta-

Snow666

ção da distribuição ISC e exemplos de arquivos em **/usr/doc/bind-9.1.0**. O SuSE os abriga em **/usr/share/doc/packages/bind8**. O SuSE inclui documentação DNS do projeto de documentação do Linux em

`/usr/share/doc/howtoen/html/DNS- O TO.html.`

O Debian fornece um exemplo de arquivo de configuração **named.conf** e de arquivos de zonas para o host local no diretório **/etc/bind**. O Debian configura o diretório-padrão do BIND em **/var/cache/bind**; assim como vem de fábrica, o diretório existe, contudo está vazio. O modelo de arquivo **named.conf** do Debian não precisa ser modificado caso você queira usar um servidor somente para caching. Se você for autorizado para algumas zonas, terá de adicioná-las ao arquivo **named.conf** fornecido.

O diretório **/usr/share/doc/bind** contém várias referências úteis. Verifique o arquivo **README.Debian** para entender a estratégia do Debian para configuração do BIND. O DNS-HOWTO pode ser encontrado no Debian no diretório **/usr/share/doc/HOWTO** com a instalação de umas variantes do pacote doc-linux (html, txt), disponível em vários idiomas.

16.17 Leitura Recomendada

O DNS e o BIND são descritos por uma série de fontes, inclusive pela documentação que acompanha a distribuição, capítulos em diversos livros sobre tópicos da Internet, um livro completo na série Nutshell da O'Reilly e vários recursos on-line.

Mailing lists e grupos de notícias

Os mailing lists a seguir estão associados ao BIND:

- bind-announce – envie um e-mail para bind-announce-request@isc.org
- namedroppers – envie um e-mail para namedroppers-request@internic.net e

especificamente para BIND 8:

- bind-users – envie um e-mail para bind-users-request@isc.org para se associar
- bind-workers – envie um e-mail para bind-workers-request@isc.org (para guerreiros da codificação)

e para BIND 9:

- bind9-users – envie um e-mail para bind-users-request@isc.org para se associar
- bind9-workers – envie um e-mail para bind-workers-request@isc.org (para guerreiros da codificação)

Envie relatos de bugs para bind-bugs@isc.org ou bind9-bugs@isc.org.

Livros e documentação diversa

THE NOMINUM BIND DEVELOPMENT TEAM. *BINDv9 Administrator Reference Manual*. Disponível na distribuição do BIND (**doc/arm**) no site www.isc.org.
Este documento descreve a administração e gerenciamento do BIND 9. Um documento anterior, *Bind Operations Guide*, ou BOG como ele é chamado, descreve detalhadamente a operação e a configuração do BIND 4. O BOG é incluído nas distribuições BIND até a versão 8.
ALBITZ, PAUL, AND CRICKET LIU. *DNS and BIND, 4. ed.*. Sebastopol, CA: O'Reilly, 2001.
Este livro popular e bem respeitado sobre BIND abrange tanto o BIND 8 quanto o BIND 9, com um pouco de BIND 4. Ele é muito completo.

Recursos on-line

O DNS Resources Directory, www.dns.net/dnsrd, é um útil conjunto de recursos e ponteiros para os recursos mantidos por András Salamon.

O arquivo "Ask Mr. DNS" de Cricket Liu pode ser pesquisado e inclui um zilhão de perguntas e respostas; consulte http://www.acmebw.com/askmrdns/archive.php

O Google indexou recursos DNS em

http://directory.google.com/Top/Computers/Internet/Protocols/DNS

Snow666

O FAQ comp.sys.tcp-ip.domains inclui um bocado de informações sobre o BIND, em grande parte sobre o BIND 4. Ele é mantido por Chris Peckham e se encontra disponível em

http://www.intac.com/~cdp/cptd-faq

O grupo de notícias comp.protocols.dns.bind também contém excelentes informações.

As RFCs

As RFCs que definem o sistema DNS estão disponíveis em www.rfc-editor.org. Idéias iniciais e em evolução aparecem primeiro na série Internet-Drafts e posteriormente vão para a série RFC. Um subconjunto selecionado de RFCs, incluindo aquelas que fizeram com que o BIND 9 fosse um grande desafio, é listado a seguir.

Os padrões definitivos e originais

- 1034 – Domain Names: Concepts and Facilities (Nomes de Domínio: Conceitos e Recursos)
- 1035 – Domain Names: Implementation and Specification (Nomes de Domínio: Implementação e Especificação)

Padrões propostos

- 1995 – Incremental Zone Transfer in DNS (Transferências Incrementais de Zona no DNS)
- 1996 – A Mechanism for Prompt Notification of Zone Changes (Um Mecanismo para a Notificação Pronta de Mudanças de Zona)
- 2136 – Dynamic Updates in the Domain Name System (Atualizações Dinâmicas no Sistema de Nomes de Domínio – DNS)
- 2181 – Clarifications to the DNS Specification (Esclarecimentos à Especificação do DNS)
- 2308 – Negative Caching of DNS Queries (Caching Negativo de Consultas DNS)

RFCs que estão se candidatando a novos padrões

- 2535 – Domain Name System Security Extensions (Extensões de Segurança ao Sistema de Nomes de Domínio – DNS)
- 2671 – Extension Mechanisms for DNS (EDNS0) (Mecanismos de Extensão para o DNS - EDNS0)
- 2672 – Non-Terminal DNS Name Redirection (DNAME) (Redireção Não Terminal de Nomes no DNS - DNAME)
- 2673 – Binary Labels in the Domain Name System (Rótulos Binários no Sistema de Nomes de Domínio)

RFCs variadas

- 1535 – A Security Problem … with Widely Deployed DNS Software (Um Problema de Segurança ... com Software DNS Amplamente Empregado)
- 1536 – Common DNS Implementation Errors and Suggested Fixes (Erros Comuns na Implementação de DNS e Correções Sugeridas)
- 1982 – Serial Number Arithmetic (Aritmética de Números Seriais)
- 2536–2541, 3007 – Various DNSSEC RFCs (Várias RFCs de DNSSEC)

Tipos de registros de recursos

- 1183 – New DNS RR Definitions: AFSDB, RP, X25, ISDN, RT (Novas Definições DNS RR: AFSDB, RP, X25, ISDN, RT)
- 1706 – DNS NSAP Resource Records (Registros de Recursos DNS NSAP)
- 1876 – A Means for Expressing Location Information in DNS (Uma Maneira para Expressar Informação Local no DNS)
- 2052 – A DNS RR for Specifying the Location of Services (SRV) (Um RR DNS para Especificação de Localização de Serviços - SRV)
- 2168 – Resolution of Uniform Resource Identifiers using DNS (Resolução de Identificadores Uniformes de Recursos URI Usando DNS)
- 2230 – Key Exchange Delegation Record for the DNS (Registro de Delegação de Intercâmbio de Chaves para o DNS)

Snow666

DNS e a Internet

- 1101 – DNS Encoding of Network Names and Other Types (Codificação DNS de Nomes de Redes e Outros Tipos)
- 1123 – Requirements for Internet Hosts: Application and Support (Requisitos para Hosts Internet: Aplicativo e Suporte)
- 1591 – Domain Name System Structure and Delegation (Estrutura e Delegação do Sistema de Nomes de Domínio)
- 2317 – Classless in-addr.arpa Delegation (Delegação sem Classe in-addr.arpa)

Operações com DNS

- 1537 – Common DNS Data File Configuration Errors (Erros de Configuração Comuns no Arquivo de Dados do DNS)
- 1912 – Common DNS Operational and Configuration Errors (Erros de Configuração Comuns na Operação do DNS)
- 2182 – Selection and Operation of Secondary DNS Servers (Seleção e Operação de Servidores DNS Secundários)
- 2219 – Use of DNS Aliases for Network Services (Uso de DNS Alternativos para Serviços de Rede)

Outras RFCs relacionadas com o DNS

- 1464 – Using DNS to Store Arbitrary String Attributes (Usando DNS para Armazenar Atributos de String)
- 1713 – Tools for DNS Debugging (Ferramentas para Depurar o DNS)
- 1794 – DNS Support for Load Balancing (Suporte do DNS para Balanceamento de Carga)
- 2240 – A Legal Basis for Domain Name Allocation (Uma Base Legal para a Alocação de Nomes de Domínios)
- 2345 – Domain Names and Company Name Retrieval (Nomes de Domínios e Recuperação de Nomes de Companhias)
- 2352 – A Convention for Using Legal Names as Domain Names (Uma Convenção para Utilização de Nomes Legais como Nomes de Domínios)

16.18 EXERCÍCIOS

E16.1 O que é DNS dividido? Por que ele é útil? O seu site está usando-o?

E16.2 Explique a função de cada um dos registros DNS: SOA, PTR, A, MX e CNAME.

E16.3 Que são registros de adesão e por que eles são necessários? Use **dig** para descobrir os registros de adesão que conectam sua zona local ao seu pai.

E16.4 Por que os servidores de nomes em **/etc/resolv.conf** são especificados com endereços IP em vez de nomes?

E16.5 Quais são as implicações do caching negativo? Por que ele é importante?

★ **E16.6** Que etapas são necessárias para configurar um novo domínio de segundo nível? Inclua tanto fatores técnicos quanto de procedimento.

★ **E16.7** Qual é a diferença entre uma resposta autorizada e não autorizada a uma consulta DNS? Como você poderia garantir que uma resposta fosse autorizada?

★ **E16.8** Que máquina é o seu servidor de nomes local? Que medidas ela deve tomar para descobrir o nome www.admin.com, supondo-se que nenhuma informação sobre este domínio seja colocada em cache em algum ponto no DNS?

★ **E16.9** O que é uma delegação defeituosa? Use **dig** para determinar se há alguma delegação defeituosa em seu site. Certifique-se de consultar de fora do site, iniciando nos servidores-raiz, de modo que você veja aquilo que é visto pelo resto do mundo.

★ **E16.10** Explique a relevância para o DNS do limite de 512 bytes dos pacotes UDP. Que subterfúgios são usados para corrigir problemas em potencial?

★★ **E16.11** Explore a implementação DNS de seu site.

a) Qual a versão do BIND que está sendo usada?

b) Há algum encaminhador configurado?

c) Que máquinas são os servidores mestre e escravo?

www.snow666.blogspot.com
Snow666

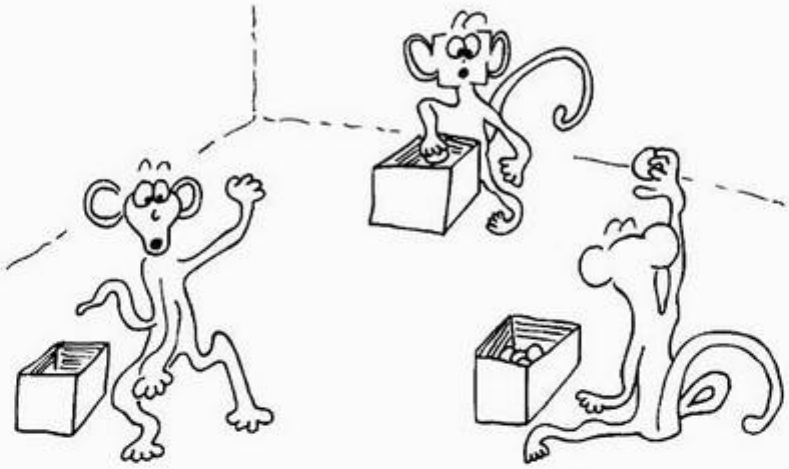
d) Elas se encontram em redes diferentes?
e) Onde são mantidos os caches?
f) Desenhe um mapa de rede do sistema.

★★ **E16.12** Copie o arquivo de zonas para o seu site local e assine-o usando as ferramentas DNSSEC distribuídas com o BIND 8 ou 9. Quanto tempo leva o processo de sinalização?
Altere um registro e assine novamente a zona — quanto tempo isto leva? Compare o tamanho "antes" e "depois" das zonas assinadas e não assinadas.
Usando **dig** para ver o que está sendo realmente retornado num pacote de resposta DNS, determine o tamanho de um pacote de retorno correspondente de uma zona assinada e não assinada. O limite UDP de 512 bytes é uma interrupção para o emprego do DNSSEC?

★★★★★ **E16.13** Executar repetidas consultas **dig** num servidor de nomes lhe mostrará como a implementação manipula questões como caching e expiração de registros.
Construa uma ferramenta que determina a versão do software de servidor de nomes que está em execução num servidor. Onde possível, use **tcpdump** para ver consultas que realmente saiam e reveja todo o tráfego, não somente o que **dig** relata. Opções **dig** como **+norecurse** podem ser úteis.
Tente obter perfis para as diversas versões do servidor de nomes Microsoft (NT, 2000, XP). Use **nmap** para confirmar suas suspeitas caso você não consiga perguntar a um administrador de sistemas local sobre a arquitetura e software executando numa determinada máquina.
nmap é considerado não amigável por vários sites; você poderia chamar sua máquina de medidas-dns.*seu.dominio* enquanto você está trabalhando neste projeto.

Snow666

17 O Sistema de Arquivos de Rede



O Network File System, comumente conhecido como NFS, permite que você compartilhe sistemas de arquivos entre computadores. O NFS é quase transparente para os usuários e é "sem estado", significando que nenhuma informação é perdida quando um servidor NFS trava. Os clientes podem simplesmente aguardar o servidor retornar e depois continuar como se nada tivesse acontecido.

O NFS foi introduzido em 1985 pela Sun Microsystems. Ele era originalmente implementado como um sistema de arquivos substituto para máquinas-cliente sem disco, porém o protocolo demonstrou ser bem projetado e muito útil como uma solução genérica para compartilhamento de arquivos. De fato, é difícil lembrar de como era a vida antes do NFS. Praticamente todas as distribuições Linux modernas possuem pelo menos um suporte mínimo ao NFS.

17.1 Informações Genéricas sobre o NFS

O NFS é formado por uma série de componentes, inclusive um protocolo de montagem e um servidor de montagem, daemons que coordenam o serviço de arquivos básico e vários utilitários para diagnóstico. Parte do software no lado servidor, bem como no lado cliente, reside no kernel. Porém, essas partes do NFS não precisam de nenhuma configuração e são amplamente transparentes sob o ponto de vista do administrador.

Versões do protocolo NFS

O protocolo NFS tem sido admiravelmente estável ao longo do tempo. Seu lançamento público original foi com a versão 2. No início dos anos 90, um conjunto de mudanças foi integrado ao protocolo para produzir a versão 3, que aumentou o seu desempenho e forneceu melhor suporte para arquivos grandes.

Já que os clientes NFS versão 2 não podem supor que uma operação de gravação esteja completa até eles receberem uma confirmação do servidor, servidores versão 2 têm de gravar cada bloco modificado para o disco antes de responder para evitar discrepâncias no caso de uma pane no sistema. Essa restrição introduz um atraso significativo nas gravações NFS, porque blocos modificados normalmente seriam gravados somente no cache de buffer em memória.

O NFS versão 3 elimina esse gargalo com um esquema de coerência que torna as gravações assíncronas seguras. Ele também atualiza vários outros aspectos do protocolo que causaram problemas de desempenho. O resultado líquido é que o NFS versão 3 é bem mais rápido do que a versão 2.

O software versão 3 é sempre capaz de interoperar com a versão 2, embora ele simplesmente reutilize a usar o protocolo anterior. O Linux tem suporte ao NFS versão 2 desde a série de kernel 1.2. O NFS versão 3 não foi introduzido até

Snow666

o kernel 2.2.18. Outras melhorias de interoperabilidade e suporte de bloqueio de arquivos também foram acrescentadas neste ponto, de modo que um kernel versão 2.2.18 ou acima é altamente recomendado caso você use o NFS.

O NFS versão 4 ainda se encontra sob desenvolvimento e ainda não é suportado pelo Linux.

Escolha de transporte

O NFS roda sobre o protocolo RPC (Remote Procedure Call) da Sun, que define uma maneira independente do sistema para os processos se comunicarem através de uma rede. Um efeito colateral vantajoso dessa arquitetura é o fato de ser possível usar tanto o UDP quanto o TCP como protocolo de transporte subjacente.

O NFS utilizava originalmente o UDP, pois este era o que apresentava melhor desempenho nas LANs e computadores dos anos 80. Embora o NFS faça a sua própria remontagem de seqüência de pacotes e verificação de erros, faltam no UDP e no NFS os algoritmos de controle de congestionamento que são essenciais para um bom desempenho numa grande rede IP.

Para remediar esses problemas em potencial, a maioria dos sistemas UNIX agora permite que você use o TCP em vez do UDP como transporte para o NFS. Essa opção foi explorada pela primeira vez como uma maneira de auxiliar o NFS a trabalhar por meio de roteadores e através da Internet. Entretanto, o consenso atual parece ser o de que o TCP normalmente é a melhor opção para tráfego NFS local também. Ao longo do tempo, a maioria das razões originais da preferência do UDP em relação ao TCP foi se evaporando na quente luz das CPUs rápidas, das memórias baratas e de controladores de rede mais inteligentes.

Grande parte dos servidores que suporta o TCP geralmente aceitará conexões em ambos os protocolos de transporte, de modo que a escolha entre TCP e UDP é tomada pelo cliente. O Linux suporta NFS sobre TCP no lado cliente, mas quando da escrita deste livro, o suporte TCP no servidor era considerado experimental ou inexistente como um todo. Você deveria considerar a possibilidade de utilizar TCPNFS numa rede experimental e ficar de olho nas experiências de terceiros antes de colocá-lo em produção.

Bloqueio de arquivos

O bloqueio de arquivos (conforme fornecido pelas chamadas de sistemas **flock** e/ou **lockf**) tem sido uma chaga nos sistemas UNIX há um longo tempo. Em sistemas de arquivos locais, é fato conhecido que ele não funciona de maneira perfeita. No contexto do NFS, o terreno é ainda mais instável. Por projeto, os servidores NFS não apresentam informações de estado: eles não têm a mínima idéia de quais máquinas estão usando um dado arquivo. Porém, essas informações são necessárias para implementar o bloqueio de arquivos. O que fazer?

A resposta tradicional tem sido implementar o bloqueio de arquivos separadamente do NFS. A maioria dos sistemas fornece dois daemons, **lockd** e **statd**, que tentam ser bem-sucedidos. Infelizmente, a tarefa é difícil por uma série de razões sutis, e o bloqueio de arquivos NFS geralmente tende a ser frágil.

Cotas de disco

O acesso a informações de cotas em discos remotos pode ser fornecido por um servidor externo, **rquotad**. Um servidor NFS impõe as cotas em disco, porém os usuários não podem visualizar suas informações de cota a menos que **rquotad** seja executado no servidor remoto. Consideramos as cotas em disco como altamente obsoletas e, portanto, não discutiremos mais **rquotad**.

Cookies e montagem sem estado

Um cliente tem de montar explicitamente um sistema de arquivos NFS antes de usá-lo, da mesma maneira que um cliente tem de montar um sistema de arquivos num disco local. Entretanto, devido ao fato do NFS ser sem estados, o servidor não controla quais clientes montaram cada sistema de arquivos. Em vez disso, o servidor apenas expõe um "cookie" secreto no final de uma negociação de montagem bem-sucedida. O cookie identifica o diretório montado para o servidor NFS, e assim fornece uma maneira para o cliente acessar seu conteúdo.

Desmontar e remontar um sistema de arquivos no servidor normalmente altera seu cookie. Como caso especial, os cookies persistem após uma reinicialização para que um servidor que trave possa retornar ao seu estado anterior. Mas não tente inicializar em modo mono usuário, mexer nos sistemas de arquivos, e depois inicializar novamente; este procedimento revogará os cookies e incapacitará os clientes de acessar os sistemas de arquivos que eles montaram até reinicializarem ou remontarem.

Assim que um cliente tiver um cookie mágico, ele usa o protocolo RPC para fazer as solicitações para operações de sistemas de arquivos, como, por exemplo criar um arquivo ou ler um bloco de dados. Devido ao NFS ser sem estados, o servidor não se importa com que solicitações o cliente fez ou não fez anteriormente. Em particular, o cliente é responsável por garantir que o servidor confirme solicitações de gravação antes de eliminar sua própria cópia dos dados a serem gravados.

www.snow666.blogspot.com
Snow666

Convenções para atribuição de nomes para sistemas de arquivos

É mais fácil gerenciar o NFS se você tiver um esquema de atribuição de nomes padrão. Os nomes que incluem o servidor (como **/anchor/tools** para um sistema de arquivos que reside em anchor) são úteis, já que eles permitem aos usuários traduzirem anúncios como "anchor estará fora do ar durante o próximo sábado por motivos de atualização" em "eu não poderei usar **/anchor/tools/TeX** no sábado para terminar a minha tese; portanto, em vez disso eu devo ir esquiar".

Infelizmente, esse esquema requer que o diretório **/anchor** exista no diretório-raiz de todas as máquinas-cliente. Se um cliente obtiver sistemas de arquivos de vários outros hosts, o raiz pode se tornar confuso. Considere a possibilidade de fornecer uma hierarquia mais profunda (por exemplo, **/home/anchor**, **/home/rastadon** etc.). Recomendamos implementar tal esquema com um dos daemons automontadores descritos na seção 17.6.

Segurança e NFS

O NFS é uma maneira conveniente para acessar arquivos em uma rede e, portanto, ele possui grande potencial para provocar problemas de segurança. Em diversos aspectos, o NFS é um "garoto-propaganda" para tudo o que há ou houve de errado com a segurança do UNIX. O protocolo foi originalmente projetado sem, fundamentalmente, nenhuma preocupação com a segurança — a conveniência tem seu preço. Felizmente, o Linux suporta uma série de recursos projetados para reduzir e isolar os problemas de segurança dos quais o NFS tradicionalmente sofria.

*Conulte a seção 17.2 para maiores informações sobre o arquivo **exports**.*

O acesso aos volumes NFS é concedido por um arquivo chamado **/etc/exports** que enumera os nomes de hosts (ou endereços IP) de sistemas que podem ter acesso aos sistemas de arquivos de um servidor. Infelizmente, essa é uma forma frágil de implementação de segurança, pois o servidor confia nos clientes para informá-lo sobre quem eles são. É fácil fazer os clientes mentirem sobre suas identidades, de modo que esse mecanismo não é totalmente confiável. Não obstante, você deve somente exportar sistemas de arquivos para clientes em que confia e deve sempre verificar se você não exportou acidentalmente sistemas de arquivos para o mundo inteiro.

O pacote de envoltório TCP pode ajudar a limitar os hosts que podem acessar sistemas de arquivos NFS. Recomendamos que você negue o acesso ao serviço **portmap** para todos que editarem **/etc/hosts.deny**; habilite o acesso a hosts e sub-redes confiáveis no arquivo **/etc/hosts.allow**. Você jamais deve colocar nada, exceto endereços IP, nas linhas `portmap` desses arquivos, pois as próprias pesquisas de nome de host poderão exigir acesso ao daemon **portmap**, resultando num loop.

Como acontece nos sistemas de arquivos locais, o controle de acesso em nível de arquivo em sistemas de arquivos NFS é controlado de acordo com o UID, o GID e as permissões do arquivo. Porém, mais uma vez, o servidor NFS confia no cliente para informá-lo quem está acessando arquivos. Se mary e bob compartilham o mesmo UID em dois clientes distintos, eles terão acesso aos arquivos NFS um do outro. Além disso, os usuários que possuem acesso root num sistema podem mudar para o UID que bem entenderem; o servidor inocentemente dará acesso aos arquivos correspondentes. Por essas razões, recomendamos enfaticamente o emprego de UIDs globalmente exclusivos e a opção `root_squash` descrita na próxima seção.

Se seu site tiver instalado um firewall de rede, é uma boa idéia bloquear o acesso às portas TCP e UDP 2049, que são usadas pelo NFS. Você também deve bloquear o acesso ao daemon **portmap**, que normalmente escuta nas portas TCP e UDP 111. Está implícito nessas precauções, porém talvez valha a pena dizer explicitamente que os sistemas de arquivos NFS não devem ser exportados para máquinas não locais.

Acesso root e a conta nobody

Embora os usuários geralmente devam receber privilégios idênticos seja para onde forem, é tradicional impedir que root seja executado irrestritamente em sistemas de arquivos montados com NFS. Como padrão, o servidor Linux NFS intercepta solicitações que chegam feitas em nome de UID 0 e as altera para que pareçam provir de algum outro usuário. Essa modificação é chamada de "squashing root" (encurralando o root). A conta root não é fechada inteiramente, contudo ela é limitada às habilidades de um usuário comum.

Uma conta substituta chamada "nobody" é definida especificamente para o pseudo-usuário como aquele que o root remoto mascara num servidor NFS. O UID tradicional de nobody é 65534 (o equivalente em complemento de dois do UID -2).[1] Você pode alterar as associações UID e GID padrão para root por meio das opções para exportação `anonuid` e `anongid`. Pode também usar a opção `all_squash` para associar os UIDs de todos os clientes ao mesmo UID no servidor. Essa configuração elimina todas as distinções entre usuários e cria uma espécie de sistema de arquivos de acesso público. Na outra ponta do espectro, a opção `no_root_squash` desativa a associação UID para root.

1. Embora o servidor NFS do Red Hat tenha como padrão UID -2, a conta nobody no arquivo **passwd** usa UID 99. Você pode ignorar isso, acrescentar uma entrada **passwd** para UID -2 ou então alterar `anonuid` e `anongid` para 99 caso queira. As demais distribuições usam UID -2 para as contas nobody e nogroup no arquivo **passwd**, como era de esperar.

Snow666

Esta opção algumas vezes é necessária para suporte a máquias-cliente sem disco ou para software que requeira acesso root ao sistema de arquivos. Geralmente é uma má idéia desativar esse recurso, porém, de qualquer maneira, a opção está lá à disposição.

O intuito por trás dessas precauções é bom, mas seu valor final não é tão bom quanto possa parecer. Lembre-se de que root num cliente NFS pode executar o comando **su** indo para o UID que ele bem entender, de modo que os arquivos de usuário jamais estarão realmente protegidos. Logins de sistema como "bin" e "sys" não são associados por UID; quaisquer arquivos que eles possuam, como. ocasionalmente, os binários do sistema ou aplicativos de terceiros, podem ser vulneráveis a ataques. O único efeito real da associação UID é o de impedir acesso a arquivos cujo proprietário seja root e que não possam ser lidos ou gravados por todo o mundo.

17.2 NFS NO LADO SERVIDOR

Diz-se que um servidor "exporta" um diretório quando ele torna este diretório disponível para uso de outras máquinas.

O processo utilizado por clientes para montar um sistema de arquivos (isto é, para tomar conhecimento do seu cookie secreto) é completamente separado do processo usado para acessar arquivos. As operações empregam protocolos distintos e as solicitações são atendidas por diferentes daemons: **mountd** para solicitações de montagem e **nfsd** para serviço de arquivos em si. Esses daemons são, na verdade, chamados **rpc.nfsd** e **rpc.mountd** como um lembrete de que eles dependem do RPC como protocolo subjacente (e, portanto, requerem que **portmap** esteja em execução; veja a seção 28.3). Omitiremos o prefixo **rpc** para fins de legibilidade.

Num servidor NFS, tanto **mountd** quanto **nfsd** devem iniciar quando o sistema inicializa, e ambos devem permanecer em execução enquanto o sistema estiver ligado. Os scripts de inicialização de sistema normalmente executarão os daemons para você se você tiver alguma exportação configurada. Os nomes dos scripts de inicialização de servidor NFS para cada plataforma são mostrados na Tabela 17.1.

mountd e **nfsd** compartilham um único banco de dados de controle de acesso que informa quais sistemas de arquivos devem ser exportados e quais clientes podem montá-los. A cópia operacional desse banco de dados normalmente é mantida num arquivo chamado **/usr/lib/nfs/xtab** além das tabelas internas ao kernel. Já que **xtab** não se destina a ser legível pelos seres humanos, usamos um comando auxiliar para adicionar e modificar entradas: **exportfs**. Para remover entradas da tabela de exportação, use **exportfs -u**.

Tabela 17.1 Scripts de inicialização de servidor NFS.

Distribuição	Caminhos para os scripts de inicialização
Red Hat	**/etc/init.d/nfs**
SuSE	**/etc/init.d/nfsserver**[a]
Debian	**/etc/init.d/nfs-kernel-server**
	/etc/init.d/nfs-common

a. **/etc/init.d/nfs** monta os sistemas de arquivos NFS de clientes no SuSE.

Na maioria dos sistemas, **/etc/exports** é a lista canônica e legível por nós seres humanos dos diretórios exportados. Como padrão, todos os sistemas de arquivos em **/etc/exports** são exportados no momento da inicialização. Você pode exportar manualmente todos os sistemas de arquivos listados em **/etc/exports** usando **exportfs -a**, que deve ser executado após o arquivo **exportfs** sofrer alterações. Também é possível exportar sistemas de arquivos apenas uma vez especificando-se o cliente, caminho e opções diretamente na linha do comando **exportfs**.

O NFS lida com a camada lógica do sistema de arquivos. Qualquer diretório pode ser exportado; ele não precisa ser um ponto de montagem ou o raiz do sistema de arquivos físico. Porém, por questões de segurança, o NFS realmente presta atenção aos limites entre os sistemas de arquivos e requer efetivamente que cada dispositivo seja exportado separadamente. Por exemplo, em uma máquina que usa uma partição **/users**, você poderia exportar o diretório-raiz sem exportar **/users**.

Os clientes normalmente têm permissão para montar subdiretórios de um diretório exportado caso eles queiram, embora o protocolo não exija este recurso. Se, por exemplo, um servidor exporta **/chimchim/users**, um cliente poderia montar somente **/chimchim/users/joe** e ignorar o restante do diretório **users**. A maioria das versões do UNIX não permite que exportemos subdiretórios de um diretório exportado com opções diferentes, porém o Linux é uma exceção.

Snow666

O arquivo exports

No arquivo **exports** do Linux, os clientes que podem vir a ter acesso a um dado sistema de arquivos são apresentados numa lista separada por espaços em branco. Cada cliente é seguido imediatamente por uma lista de opções separadas por vírgulas e entre parênteses. As linhas podem ser continuadas com uma barra invertida.

Eis o aspecto deste formato:

```
/home/boggs          inura(rw,no_root_squash) lappie(rw)
/usr/share/man       *.toadranch.com(ro)
```

Não há nenhum modo de listar vários clientes num único conjunto de opções, embora algumas especificações de clientes realmente se refiram a vários hosts. A Tabela 17.2 lista os quatro tipos de especificações de clientes que podem aparecer no arquivo **exports**.

A Tabela 17.3 mostra as opções de exportação mais comuns.

O servidor NFS do Linux possui o recurso incomum de permitir que subdiretórios de diretórios exportados sejam exportados com opções diferentes. A opção `noaccess` é fornecida para "anular a exportação" de subdiretórios que você não quer compartilhar.

Tabela 17.2 Especificações de clientes no arquivo /etc/exports.

Tipo	Sintaxe	Significado
Nome de host	*hostname*	Hosts individuais
Grupo de redes	@ *groupname*	Grupos de rede NIS; consulte a página 372 para obter maiores detalhes
Curingas	* e ?	FQDNs[a] com curingas. "*" Não coincide com um ponto.
Redes IP	*endereçoIP/máscara*	Especificações tipo CIDR (por exemplo, 128.138.92.128/25)

a. Nomes de domínios totalmente qualificados.

Tabela 17.3 Opções comuns para exportação.

Opção	Descrição
`ro`	Exporta somente leitura
`rw`	Exporta para leitura e gravação (o padrão)
`rw=` *`lista`*	Exporta na maior parte somente para leitura. *lista* enumera os hosts que têm permissão para montar para gravação; todos os demais têm de montar somente para leitura.
`root_squash`	Associa ("squashes", ou seja, "encurrala") UID 0 e GID 0 aos valores especificados por `anonuid` e `anongid`. Este é o padrão.
`no_root_squash`	Permite acesso normal por parte de root. Perigoso.
`all_squash`	Associa todos os UIDs e GIDs a suas versões anônimas. Útil para suportar PCs e hosts monousuários não confiáveis.
`anonuid=` *xxx*	Especifica o UID para os quais roots remotos devem ser encurralados
`anongid=` *xxx*	Especifica o GID para os quais roots remotos devem ser encurralados
`secure`	Requer que o acesso remoto se origine de uma porta privilegiada
`insecure`	Permite acesso remoto de qualquer porta
`noaccess`	Impede o acesso a este diretório e seus subdiretórios (usado com exportações aninhadas)

Por exemplo, a configuração

```
/home          *.toadranch.com(rw)
/home/boggs    (noaccess)
```

permite que hosts no domínio toadranch.com acessem todo o conteúdo de **/home**, exceto **/home/boggs**. A ausência de um nome de cliente na segunda linha significa que a opção se aplica a todos os hosts; talvez seja um pouco mais seguro proceder desta maneira.

Não se esqueça de executar **exportfs -a** após atualizar o arquivo **exports** para fazer com que suas alterações tenham efeito.

Snow666

nfsd: servir arquivos

Assim que a solicitação de montagem de um cliente tiver sido validada por **mountd**, o cliente poderá solicitar várias operações de sistemas de arquivos. Essas solicitações são manipuladas no servidor por **nfsd**, o daemon de operações NFS.[2] **nfsd** não precisa ser executado num cliente NFS a menos que o cliente exporte sistemas de arquivos próprios.

nfsd recebe um argumento numérico que especifica quantos threads de servidor deve iniciar. Selecionar o número apropriado de **nfsd**s é importante e, infelizmente, envolve uma certa dose de magia. Se o número for muito baixo ou muito alto, o desempenho do NFS pode sofrer com isto.

O desempenho de sistemas mais antigos poderia degradar bastante como conseqüência de ter-se muitos **nfsd**s, pois o kernel despertaria todos os **nfsd**s ociosos em resposta a cada solicitação que chega. Os sistemas Linux se comportam melhor em relação a isso, pois eles usam threads em vez de processos separados, de modo que é normal transgredir um pouco em nome da generosidade.

Em termos genéricos, 8 threads **nfsd** é adequado para um servidor utilizado com pouca freqüência e é suficientemente pouco para que não surjão problemas de desempenho. Em um servidor num ambiente de produção, algo entre 12 e 20 é um bom número. Se perceber que a média da carga (conforme relatado por **uptime**) estiver aumentando à medida que acrescenta **nfsd**s, você acabou indo longe demais; retorne um pouco a partir desse limiar. Você também deve executar **nfsstat** regularmente para verificar problemas de desempenho que poderiam estar associados ao número de threads **nfsd**. Veja a seção 17.4 para obter maiores informações sobre **nfsstat**.

Em um servidor NFS com muitos clientes UDP, sockets UDP poderão estourar caso as solicitações cheguem enquanto todos os threads **nfsd** já estiverem em uso. Você pode monitorar o número de estouros através de **netstat -s**. Adicione mais **nfsd**s até que o número de estouros de socket UDP caia para zero. Estouros indicam uma grave falta de suprimento de daemons de servidor, de modo que você provavelmente deverá adicionar alguns a mais do que essa métrica indicar.

Pode-se alterar o numero de processos **nfsd** editando-se o script de inicialização apropriado em **/etc/init.d**. Veja a Tabela 17.1 para obter o nome do script a ser editado.

17.3 NFS NO LADO CLIENTE

Os sistemas de arquivos são montados quase que da mesma maneira que sistemas de arquivos de disco locais. O comando **mount** entende a notação *nome_do_host:diretorio* como o caminho do *diretorio* na máquina *nome_do_host*. Da mesma forma que ocorre com sistema de arquivos locais, **mount** associará o *diretorio* remoto do *host* remoto a um diretório contido na árvore de arquivos local. Após a montagem, um sistema de arquivos montado com NFS é acessado da mesma maneira que um sistema de arquivos local. O comando **mount** e suas extensões NFS associadas representam as preocupações mais significativas para um administrador de sistemas de um cliente NFS.

Antes de um sistema de arquivos NFS poder ser montado, ele deve ser exportado apropriadamente (veja a seção 17.2). Para verificar se um servidor exportou apropriadamente seus sistemas de arquivos sob a perspectiva do cliente, você pode usar o comando **showmount** do cliente:

```
% showmount -e coyote
Export list for coyote:
/home/boggs inura.toadranch.com
```

Esse exemplo indica que o diretório **/home/boggs** no servidor coyote foi exportado para o sistema-cliente inura.toadranch.com. A saída gerada por **showmount** deveria ser a primeira coisa a checar caso uma montagem nfs não esteja funcionando e se você já tiver verificado que os sistemas de arquivos foram exportados apropriadamente no servidor via **exportfs**. (Pode ser que você simplesmente tenha se esquecido de executar **exportfs -a** após atualizar o arquivo **exports**.)

Se o diretório tiver sido exportado apropriadamente no servidor, porém **showmount** retornar um erro ou uma lista vazia, talvez você tenha de verificar cuidadosamente que todos os processos necessários estejam sendo executados no servidor (**portmap**, **mountd**, **nfsd**, **statd** e **lockd**), que os arquivos **hosts.allow** e **hosts.deny** permitam acesso a estes daemons e que você se encontre no sistema-cliente correto.

Para efetivamente montar o sistema de arquivos, você usaria um comando como indicado a seguir:

```
# mount -o rw,hard,intr,bg coyote:/home/boggs /coyote/home/boggs
```

2. Na realidade, **nfsd** simplesmente faz uma chamada de sistema sem retorno ao código de servidor NFS embutido no kernel.

www.snow666.blogspot.com
Snow666

As opções após **-o** indicam que o sistema de arquivos deve ser montado para leitura/gravação, que é possível interromper as operações e que novas tentativas devem ser feitas em segundo plano. Esses flags são bastante padrões; outros flags comuns são apresentados na Tabela 17.4.

Tabela 17.4 Flags de montagem NFS.

Flag	Descrição
`rw`	Monta o sistema de arquivos para leitura-gravação (tem de ser exportado desta maneira).
`ro`	Monta o sistema de arquivos somente para leitura.
`bg`	Se a montagem falhar (o servidor não responder), continue tentando em segundo plano e continue com outras solicitações de montagem.
`hard`	Se um servidor sair do ar, as operações que tentam acessá-lo serão bloqueadas até que o servidor volte a entrar em operação.
`soft`	Se um servidor sair do ar, as operações que tentam acessá-lo falharão e retornará um erro. Este recurso é útil para evitar que processos fiquem "pendurados" em montagens não essenciais.
`intr`	Permite que os usuários interrompam operações bloqueadas (e façam com que elas retornem um erro).
`nointr`	Não permite que o usuário provoque uma interrupção.
`retrans=n`	Especifica o número de vezes para repetição de uma solicitação antes de retornar um erro num sistema de arquivos montado com a opção soft.
`timeo=n`	Configura o período de timeout (em décimos de segundo) para as solicitações.
`rsize=n`	Configura o tamanho do buffer de leitura para *n* bytes.
`wsize=n`	Configura o buffer de gravação para *n* bytes.
`nfsvers=n`	Seleciona as versões 2 ou 3 do protocolo NFS (normalmente de forma automática).
`tcp`	Seleciona o transporte via TCP. UDP é o padrão.[a]

a. Note que a maioria dos servidores NFS para Linux não suporta o TCP como protocolo de transporte.

Sistemas de arquivos montados com a opção hard (o padrão) podem fazer com que processos fiquem pendurados quando seus servidores ficarem inativos. Esse comportamento é particularmente incômodo quando os processos em questão são daemons-padrão, de modo que não recomendamos servir binários de sistema críticos via NFS. Geralmente, o emprego das opções `soft` e `intr` reduzirá o número de dores de cabeça relacionadas ao NFS. Entretanto, essas opções podem ter seus efeitos colaterais indesejáveis, como abortar uma simulação de 20 horas depois de ele ter sido executado por 18 horas simplesmente devido a um probleminha transitório de rede.[3] Soluções de automontagem como autofs e **amd**, discutidas posteriormente neste capítulo, também fornecem algumas curas para doenças de montagem.

Os tamanhos de buffer de escrita e de gravação se aplicam tanto a montagens UDP quanto TCP, porém os valores ótimos diferem. Pelo fato de se poder confiar no TCP para transferência de dados eficiente, os valores devem ser maiores; 32 K é uma boa opção. Para o UDP, os valores indicados quando o servidor e o cliente se encontram na mesma rede é de 8 K.[4] O padrão é 1 K, muito embora o manual recomende aumentá-lo para 8 K para fins de melhora desempenho.

Entretanto, nos kernels 2.2 e 2.4 do Linux, o tamanho da fila de entrada padrão é de 64 K. Com oito threads **nfsd** em execução no servidor NFS, somente uma solicitação pode estar pendente em cada instância de **nfsd** antes de as solicitações começarem a ser descartadas. Portanto, talvez se deva considerar aumentar o tamanho da fila de recebimento somente para o **nfsd**, retornando-a ao valor-padrão após executar **nfsd** para que outros processos não sejam afetados negativamente pela modificação. É possível alterar o tamanho da fila de entrada nos scripts de inicialização de sistema pelo emprego de procfs. Este exemplo configura o tamanho da fila em 256 K, que é um padrão razoável:

```
rmem_default='cat /proc/sys/net/core/rmem_default'
rmem_max='cat /proc/sys/net/core/rmem_max'
```

3. Jeff Forys, um de nossos revisores técnicos, salientou: "A maioria das montagens deveria usar `hard`, `intr` e `bg`, pois estas opções preservam melhor os objetivos de projeto originais do NFS (confiabilidade e ausência de estados). `soft` é abominável, um horrível hack satânico! Se o usuário quiser interromper, tudo bem. Caso contrário, aguarde o servidor e tudo eventualmente estará bem novamente sem nenhuma perda de dados."
4. Se você usar `ipchains` ou `iptables`, talvez tenha de adicionar uma regra para aceitar fragmentos, pois 8 K está acima do MTU para Ethernet. Aceitar fragmentos pode lhe tornar mais vulnerável a um ataque de recusa de serviços.

Snow666


```
echo 262144 > /proc/sys/net/core/rmem_default
echo 262144 > /proc/sys/net/core/rmem_max
```

Execute ou reinicialize **rpc.nfsd**, depois retorne a configuração para seus valores originais:

```
echo $rmem_default > /proc/sys/net/core/rmem_default
echo $rmem_max > /proc/sys/net/core/rmem_max
```

A montagem pode ser testada com **df** da mesma maneira como se testa um sistema de arquivos local:

```
% df /coyote/home/boggs
Filesystem            1k-blocks      Used  Available   Use%  Mounted on
coyote:/home/boggs    17212156    1694128  14643692     11%  /coyote/home/boggs
```

As partições NFS podem ser desmontadas pelo comando **umount**. Se o sistema de arquivos NFS estiver em uso ao tentar desmontá-lo, será gerado um erro como

```
umount: /coyote/home/boggs: device is busy
```

Da mesma maneira como outros sistemas de arquivos, um sistema de arquivos NFS não pode ser desmontado enquanto estiver sendo usado. Use **lsof** para localizar processos com arquivos abertos no sistema de arquivos; interrompa-os ou mude diretórios no caso de shells. Se todas essas medidas falharem ou se seu servidor estiver inativo, tente **umount -f** para forçar o sistema de arquivos a ser desmontado.

*Consulte a seção 17.6 para obter maiores informações sobre autofs e **amd**.*

Montando sistemas de arquivos remotos no momento da inicialização

Você pode usar o comando **mount** para estabelecer montagens de rede temporárias, mas deve listar montagens que fazem parte de uma configuração permanente do sistema em **/etc/fstab** para que elas sejam montadas automaticamente no momento da inicialização. De forma alternativa, as montagens podem ser manipuladas por um serviço de montagem automática como **automount** ou **amd**; veja a seção 17.6.

As entradas **fstab** a seguir montam os sistemas de arquivos **/home** e **/usr/local** por meio dos hosts coyote e inura:

```
# sistema_de_arquivos   ponto_de_montagem  tipo_do_sistema_de_arquivos  flags            dump  fsck
coyote:/home            /coyote/home       nfs                          rw,bg,intr,hard  0     0
inura:/usr/local        /usr/local         nfs                          ro,bg,intr,soft  0     0
```

*Consulte à página 107 para obter maiores informações sobre o arquivo **fstab**.*

Ao adicionar entradas a **fstab**, certifique-se de criar os diretórios de ponto de montagem apropriados com **mkdir**. É possível fazer que as alterações passem a ter efeito imediatamente (sem ter de reinicializar o sistema) executando **mount -a -t nfs** para montar todos os sistemas de arquivos do tipo `nfs` em **fstab**.

O campo `flags` de **/etc/fstab** especifica opções para montagens NFS; essas opções são as mesmas que especificaríamos na linha de comando **mount**.

Restrições de portas seguras

Clientes NFS são livres para usar qualquer porta de origem TCP ou UDP que eles quiserem ao se conectarem a um servidor NFS. Entretanto, os servidores Linux podem insistir que as solicitações provenham de uma porta privilegiada (uma porta com numeração inferior a 1024) se o sistema de arquivos for exportado com a opção de exportação `secure`, que é ativa como padrão. No mundo dos PCs e desktops Linux, o emprego de portas privilegiadas fornece na verdade pouca segurança.

Os clientes NFS Linux adotam a abordagem tradicional (e ainda recomendada) de usar como padrão uma porta privilegiada, para impedir potencial para conflito. Para aceitar montagens de portas de origem não privilegiadas, exporte o sistema de arquivos com a opção de exportação `insecure`.

17.4 NFSSTAT: FAZENDO O DUMP DE ESTATÍSTICAS NFS

O Linux dispõe de um comando chamado **nfsstat** capaz de exibir várias informações estatísticas mantidas pelo sistema NFS. **nfsstat -s** exibe estatísticas para processos de servidor NFS, e **nfsstat -c** mostra informações relativas a operações no cliente.[5] Por exemplo:

5. Na época em que este livro era escrito, a página de manual para `nfsstat(8)` afirmava que o cliente NFS ainda não registrava informações estatísticas em log. Porém, a execução de nfsstat realmente relata informações estatísticas de cliente. O seu caso pode variar.

Snow666


```
% nfsstat -s
Server rpc stats:
calls        badcalls        badauth         badclnt         xdrcall
24314112     311             9               302             0
Server nfs v2:
getattr      null            setattr         root            lookup          readlink
8470054  34% 58      0%      55199    0%     0        0%     1182897  4%     917      0%
read         wrcache         link            create          remove          rename
6602409  27% 0       0%      7452     0%     61544    0%     46712    0%     11537    0%
write        symlink         mkdir           rmdir           readdir         fsstat
7785789  32% 744     0%      3446     0%     2539     0%     13614    0%     69201    0%
```

Esse exemplo é de um servidor NFS relativamente saudável. Se mais de 3% das chamadas forem ruins, é provável que haja um problema com o seu servidor NFS ou com a rede. Verifique a saída gerada por **netstat -s** para estatísticas de rede genéricas. Ela poderá revelar problemas com pacotes perdidos, remontagem de fragmentos ou estouros de fila de rede que afetarão o desempenho NFS. Veja a seção 20.4 para mais informações sobre como depurar uma rede com **netstat**.

Executar **nfsstat** e **netstat** ocasionalmente e se familiarizar com sua saída o ajudará a descobrir problemas de NFS antes de seus usuários o fazerem.

17.5 Servidores de Arquivos NFS Dedicados

Serviço de arquivos rápido e confiável é um dos elementos de maior importância de qualquer ambiente computacional de produção. Embora você possa certamente rodar o seu próprio servidor de arquivos em uma estação de trabalho Linux e um punhado de discos rígidos comuns, fazer isto normalmente não é a melhor solução em termos de desempenho e de facilidade de administração (embora ela normalmente seja a mais barata).

Servidores de arquivos NFS dedicados já estão no mercado aproximadamente há mais de uma década. Eles oferecem uma infinidade de vantagens em potencial em relação à abordagem caseira:

- São otimizados para serviços de arquivos e fornecem tipicamente o melhor desempenho NFS possível.
- À medida que as exigências de armazenamento aumentam, eles podem ser ampliados de maneira gradual para suportar terabytes de armazenamento e centenas de usuários.
- São mais confiáveis do que os computadores Linux graças a seus softwares simplificados, hardware redundante e uso de espelhamentos de discos.
- Normalmente fornecem serviços de arquivos tanto para clientes Linux quanto clientes Windows. Alguns até mesmo contêm servidores FTP e Web completos.
- Eles normalmente são mais fáceis de serem administrados do que os servidores de arquivos Linux.
- Normalmente fornecem recursos para backup e verificação que são superiores àqueles encontrados em sistemas Linux comuns.

Das ofertas atuais, nossas favoritas são aquelas produzidas pela Network Appliance, Inc. (www.netapp.com). Seus servidores abrangem uma gama que vai desde os muito pequenos até os muito grandes e seus preços são razoáveis. Auspex e EMC são fabricantes no mercado de servidores topo de linha. Eles oferecem bons produtos, porém esteja preparado para o choque do preço e prepare sua tolerância para chavões de marketing.[6]

17.6 Montagem Automática

Montar sistemas de arquivos um a um listando-os em **/etc/fstab** introduz uma série de problemas em redes grandes.

Primeiramente, manter **/etc/fstab** numa centena de máquinas pode ser tedioso. Cada uma pode ser ligeiramente diferente e, portanto, requerer atenção individual.

Em segundo lugar, se os sistemas de arquivos forem montados de vários hosts diferentes, o caos se estabelecerá quando um destes servidores travar. Cada comando que acessar os pontos de montagem ficará pendurado.

Em terceiro lugar, quando um servidor trava, ele pode incapacitar usuários tornando indisponíveis partições importantes como **/usr/share/man**. Nessa situação, é melhor se uma cópia da partição puder ser montada temporariamente em um servidor de backup.

6. Falando em chavões, um dos principais que você ouvirá neste contexto é "armazenamento anexável à rede", também conhecido como NAS. É somente uma maneira extravagante de dizer "serviço de arquivos".

www.snow666.blogspot.com
Snow666

Um daemon de automontagem monta sistemas de arquivos quando são referidos e os desmonta quando eles não são mais necessários. Esse procedimento minimiza o número de pontos de montagem ativos e é em grande parte transparente para o usuário. Com a maioria dos automontadores, também é possível fornecer uma lista de sistemas de arquivos "replicados" (idênticos) para que a rede possa continuar a funcionar quando um servidor primário se tornar indisponível.

Para implementar essa montagem e desmontagem nos bastidores, o automontador monta um driver de sistemas de arquivos virtual nos diretórios que você designou como localizações para a ocorrência de montagem automática. No passado, o automontador fazia isso se apresentando como um servidor NFS, porém este esquema padece de várias limitações significativas e raramente é encontrado nos sistemas contemporâneos. Na atualidade, é usado um driver de sistema de arquivos residente em kernel chamado autofs.

Em vez de espelhar um sistema de arquivos de verdade, um automontador "falsifica" uma hierarquia de sistema de arquivo de acordo com as especificações que você lista em seu arquivo de configuração. Quando um usuário fizer referência a um diretório dentro do sistema de arquivos virtual do automontador, o automontador interceptará a referência e montará o sistema de arquivos que o usuário está tentando alcançar.

A idéia de um automontador provém originalmente da Sun. O automontador do Linux, chamado **automount**, foi desenvolvido para simular o automontador da Sun, embora ele seja uma implementação independente do conceito e seja diferente numa série de aspectos.

amd, originalmente escrito por Jan-Simon Pendry do Imperial College em Londres, é o resultado de uma tese de doutorado que expandiu a idéia original da Sun. **amd** corrigiu muitos dos problemas no automontador original da Sun. Ele também apresentava a vantagem de ser livre e poder ser instalado em uma grande variedade de sistemas UNIX, e pode ser compilado ou instalado em um pacote para ser executado no Linux.

A configuração de **automount** é relativamente simples e concisa. **amd** é mais complicado, mais poderoso e mais portável. Ele também inclui alguns recursos que são em grande parte experimentais. Talvez você queira verificar a lista de recursos de **amd** e ver se há algo com o qual não possa viver sem.

17.7 AUTOMOUNT

automount é um processo em segundo plano que configura um único ponto de montagem para autofs, a porção kernel do automontador do Linux. O script de inicialização **/etc/init.d/autofs** analisa sintaticamente um arquivo mestre (normalmente **/etc/auto.master**) e executa **automount** para cada um dos pontos de montagem listados. É típico ver uma instância em execução de **automount** para cada ponto de montagem automático que foi configurado.

Raramente você precisará executar o comando **automount** diretamente, já que quase toda a administração do automontador é realizada através do script **/etc/init.d/autofs**.[7] Como ocorre com a maioria dos scripts de inicialização, o script **autofs** aceita um único parâmetro na linha de comando que pode ser **start**, **stop**, **reload**, **restart** ou **status**. Sempre que forem feitas modificações na configuração do automontador, você terá de executar **autofs reload** para que as alterações entrem em efeito. **autofs status** lhe dá o estado de automontagens existentes.

O arquivo **auto.master** associa um ponto de montagem a um "mapa". Um mapa converte o nome de diretório acessado — conhecido como a "chave" — numa linha de comando que **mount** pode usar para realizar a montagem real. Um mapa pode ser um arquivo de texto, um programa executável ou um banco de dados NIS ou LDAP.

Quando um usuário faz referência a um diretório que foi montado com o módulo de sistema de arquivos de kernel autofs, ele notifica o processo **automount** sobre o acesso. O processo **automount** descobre qual sistema de arquivos montar consultando um arquivo ou programa de mapas, depois realiza a montagem antes de retornar o controle ao usuário que disparou a pesquisa.

É possível ver os sistemas de arquivos autofs e os processos **automount** aos quais eles estão anexados executando-se os comandos **mount** e **ps**:

```
% mount
/dev/hda3 on / type ext2 (rw)
proc on /proc type proc (rw)
/dev/hda1 on /boot type ext2 (rw)
automount(pid8359) on /misc type autofs          // sistema de arquivos do automontador
   (rw,fd=5,pgrp=8359,minproto=2,maxproto=4)
automount(pid8372) on /net type autofs           // sistema de arquivos do automontador
```

7. Não confunda o script **autofs** com o sistema de arquivos autofs. A relação entre eles é que o script informa ao kernel como configurar o sistema de arquivos.

Snow666


```
   (rw,fd=5,pgrp=8372,minproto=2,maxproto=4)

% ps auxw | grep automount
root      8359  0.0   1.0   1360  652 ?       S     Dec27     0:00
   /usr/sbin/automount /misc file /etc/auto.misc
root      8372  0.0   1.0   1360  652 ?       S     Dec27     0:00
   /usr/sbin/automount /net program /etc/auto.net
```

Aqui podemos observar dois sistemas de arquivos autofs montados em **/misc** e **/net**. Esses sistemas de arquivos virtuais são anexados aos processos **automount** com PIDs 8359 e 8372, respectivamente.

Os comandos **automount** executados pelo script **/etc/init.d/autofs** pode ser visto na saída gerada por **ps**. **auto.misc** é um arquivo de mapas comum e **auto.net** é um programa executável. Estes mapas são descritos com mais detalhes a seguir.

O arquivo mestre

O arquivo mestre lista os diretórios que deveriam ter sistemas de arquivos autofs montados neles e associa um mapa a cada diretório. Além de fornecer o diretório-raiz para o mapa e o nome do mapa, você também pode especificar opções adicionais no formato "**-o**" usado pelo comando **mount**. Essas opções se aplicarão a cada entrada do mapa. As convenções do Linux diferem daquelas da Sun em que as opções do arquivo mestre são cumulativas com aquelas do mapa; ambos os conjuntos de opções são transmitidos para o **mount**.

Um arquivo mestre simples que faz uso do arquivo de mapas mostrado na próxima seção ficaria parecido com o seguinte:

```
# Opções     de Mapas          de Diretórios
/chimchim    /etc/auto.chim    -secure,hard,bg,intr
```

O arquivo mestre pode ser substituído ou aumentado por uma versão compartilhada através do NIS. A fonte das informações de automontagem do sistema é especificada pelo campo `automount` em **/etc/nsswitch.conf**. Veja *Priorizando fontes de informações administrativas* na página 373 para obter maiores informações sobre o arquivo **nsswitch.conf**.

Arquivos de mapas

Os arquivos de mapas (conhecidos como "mapas indiretos" em outros sistemas) montam automaticamente diversos sistemas de arquivos sob um diretório comum. O caminho do diretório é especificado no arquivo mestre, não no próprio mapa. Por exemplo, um mapa para sistemas de arquivos montados sob **/chimchim** (correspondente ao exemplo acima) poderia ser semelhante ao seguinte:

```
users     chimchim:/chimchim/users
devel     -soft,nfsproto=3 chimchim:/chimchim/devel
info      -ro chimchim:/chimchim/info
```

A primeira coluna nomeia o subdiretório em que cada automontagem deve ser instalada, e os itens subseqüentes listam as opções de montagem e o caminho de origem do sistema de arquivos. Esse exemplo (armazenado em **/etc/auto.chim**) informa a **automount** que ele pode montar os diretórios **/chimchim/users**, **/chimchim/devel** e **/chimchim/info** no host chimchim, com **info** sendo montado somente para leitura e **devel** sendo montado com a opção soft com protocolo NFS versão 3.

Nessa configuração os caminhos em chimchim e no host local serão idênticos, porém a correspondência não é obrigatória.

Mapas executáveis

Se um arquivo de mapas indireto é executável, supostamente ele é um script ou programa que gera dinamicamente informações de automontagem. Em vez de ler o mapa como um arquivo de texto, o automontador o executará com um argumento (a chave) que indica qual subdiretório um usuário tentou acessar. O script é responsável por imprimir uma entrada de mapa apropriada; se a chave especificada não for válida, o script pode simplesmente sair sem imprimir nada.

Este recurso é muito poderoso e compensa as potenciais deficiências no sistema de configuração bastante estranho de **automounter**. De fato, ele permite que você defina facilmente um arquivo de configuração de automontagem válido para um site num formato de sua própria preferência. Você pode escrever um script Perl simples para decodificar a configuração global em cada máquina. Alguns sistemas vêm acompanhados com um prático mapa executável **/etc/auto.net** que utiliza um nome de host como uma chave e monta todos os sistemas de arquivos exportados neste host.

Snow666

17.8 AMD: Um Automontador Mais Sofisticado

amd é uma forma mais elaborada do conceito de automontador. Ele se tornou uma espécie de órfão quando seu autor original parou de mantê-lo. Agora ele é preservado por Erez Zadok na Universidade de Columbia como um componente do pacote **am-utils**. O URL atual é

http://www.cs.columbia.edu/~ezk/am-utils

amd oferece as seguintes vantagens em relação a **automount**:

- **amd** envia consultas "mantenha vivo" para servidores remotos em intervalos regulares e mantém uma lista de servidores que são acessíveis. **amd** irá montar, desmontar e substituir sistemas de arquivos baseado nessas informações. Se um servidor travar, futuros acessos ao sistema de arquivos retornarão um erro "operação vai bloquear" em vez de ficar pendurado.
- **amd** não contém nenhum código-fonte proprietário e foi portado para mais de 20 versões do UNIX, inclusive o Linux.
- **amd** oferece suporte para uma série de tipos de montagens, como a montagem "união", que não é suportada por **automount**.
- A distribuição de **amd** inclui uma ferramenta de consulta e manipulação, **amq**, que lhe permite monitorar o estado de **amd** e enviar dicas e comandos (como solicitações de desmontagem forçadas).
- A sintaxe de mapas de **amd** é mais genérica do que a de **automount**. Você pode criar um único arquivo para todos os hosts em seu site e distribuí-los com **rdist** ou **rsync**.
- **amd** se baseia no conceito de que cada servidor possui um ou mais sistemas de arquivos, em que cada sistema de arquivos contém um ou mais volumes (um conjunto coerente de arquivos). Isso torna a manipulação de subdiretórios mais direta do que com **automount**.
- **amd** não irá decepcioná-lo.

Mapas amd

O formato de mapas **amd** é extremamente flexível e permite que o mesmo arquivo de configuração seja usado em muitas máquinas. As entradas de mapas podem conter condicionais que as ativam somente em contextos específicos (por exemplo, num host ou tipo de máquina específico). As condicionais usam "variáveis seletoras" já embutidas que são preenchidas com vários trechos de informações sobre o ambiente no qual **amd** está sendo executado. As variáveis seletoras mais comumente usadas são apresentadas na Tabela 17.5.

Tabela 17.5 Variáveis seletoras de amd.

Variável	Descrição
`arch`	Arquitetura da máquina atual
`autodir`	Diretório-padrão sob o qual se deve montar sistemas de arquivos
`byte`	Identificação do tipo de CPU (big-endian ou little-endian)
`cluster`	Nome do cluster local das máquinas; o padrão é `domain`
`domain`	Nome de domínio NIS local
`host`	Nome de host local
`hostd`	Nome de host concatenado com o nome de domínio DNS local
`karch`	Arquitetura de kernel (o padrão é o valor do seletor `arch`)
`key`	Nome de volume que está sendo determinado
`map`	Nome do mapa de montagem que está sendo usado
`network`	Coincide com o nome da rede ou com o número de qualquer interface de rede
`os`	Sistema operacional
`osver`	Versão do sistema operacional

A entrada para um ponto de montagem pode descrever várias coisas diferentes que poderiam ser montadas lá. Por exemplo, o arquivo de mapas a seguir informa **amd** sobre dois sistemas de arquivos: **/usr/man** e **/cs/tools**. Cada

Snow666

sistema de arquivos possui um conjunto de opções que controla a montagem na máquina onde o sistema de arquivos reside efetivamente e um outro que monta o sistema de arquivos pela rede.

```
/default        opts:=rw,soft,timeo=10,retrans=5

usr/man         host==chimchim;type:=ufs;dev:=/dev/sd1f \
                host!=chimchim;rhost=chimchim;rfs:=/${key};\
                type=nfs;fs:=${autodir}/${key}
cs/tools        host==anchor;type:=ufs;dev:=/dev/sd3c \
                host!=anchor;rhost=anchor;rfs:=/${key};\
                type=nfs;fs:=${autodir}/${key}
```

Elementos do tipo name:=value definem vários atributos de **mount**. Por exemplo, a primeira linha configura as opções de montagem padrão para "`rw,soft,timeo=10,retrans=5`".

Elementos do tipo name==value ou name!=value são condicionais; elementos subseqüentes são usados somente se a condição resultar em verdadeira. Notações como `${autodir}` e `${key}` inserem o valor da variável apropriada.

A estrutura `/default` especifica padrões que se aplicam a todas as entradas e mapas a menos que elas sejam explicitamente sobrepostas. A Tabela 17.6 descreve as diversas opções.

Tabela 17.6 Opções de mapas para amd.

Opção	Descrição
`rhost`	Host remoto no qual o volume reside
`rfs`	Nome do sistema de arquivos remoto
`type`	Tipo de montagem, normalmente `nfs` ou `ufs` (disco local)
`fs`	Ponto de montagem local
`opts`	Opções de mount
`addopts`	Opções a serem acrescentadas às opções-padrão
`remopts`	Opções a serem usadas caso o servidor não seja local (por exemplo, tamanhos de leitura/gravação menores)

Iniciando amd

Uma cópia em execução de **amd** gerencia um diretório de sistema de arquivos virtual sob o qual ocorrem todas as ações relativas ao processo de automontagem. O nome do diretório virtual e o arquivo de mapas que informa o que montar dentro dele são especificados na linha de comando.

amd pode ser iniciado com um script como o mostrado a seguir:

```
#!/bin/csh -f
cd /usr/local/etc/amd
exec /usr/local/bin/amd -x fatal,error,user -r -l syslog -a /tmp_mnt
   /amd amd.master.map >& /dev/console
```

As opções usadas nesse script são descritas na Tabela 17.7.

Tabela 17.7 Opções de linha de comando do amd.

Opção	Descrição
-x	Configura opções de registro em tempo de execução
-r	"Adota" montagens existentes
-l	Especifica arquivo de log ou **syslog** para registro das mensagens de erro
-a	Especifica uma localização alternativa para pontos de montagem[a]
/amd	Configura o diretório virtual (automontagem)
amd.master.map	Especifica o arquivo de mapas contendo as opções de montagem

a. O padrão é **/a**.

Snow666

Quando um usuário faz referência a um dos sistemas de arquivos definidos no arquivo de mapas de **amd**, **amd** monta o sistema de arquivos e monitora o emprego subseqüente da montagem. Após ele estar inativo por um determinado período (normalmente de 5 a 15 minutos), **amd** desmonta o sistema de arquivos até ele ser referido novamente.

Use o comando **amq** para verificar o estado das montagens.

Interrompendo amd

O comando **amd** precisa ser interrompido de maneira graciosa para ter a chance de se desvencilhar da estrutura de sistema de arquivos. Enviar um SIGTERM a **amd** é a maneira educada de solicitar que ele termine.

17.9 Leitura Recomendada

ZADOK, EREZ. *Linux NFS and Automounter Administration*. Sybex. 2001.
Erez Zadok mantém o **amd**; seu livro aborda o **amd**, bem como o ambiente NFS no Linux.
CALLAGHAN, BRENT. *NFS Illustrated*. Addison-Wesley. 1999.
PENDRY, JAN-SIMON; NICK WILLIAMS. "AMD: The 4.4BSD Automounter Reference Manual." 4.4BSD System Manager's Manual, Usenix and O'Reilly. 1994.
STERN, HAL; MIKE EISLER; RICARDO LABIAGA. *Managing NFS and NIS*. 2. ed. Sebastopol: O'Reilly & Associates, 2001.

A Tabela 17.8 lista as diversas RFCs para o protocolo NFS.

Tabela 17.8 RFCs relacionadas com o NFS.

RFC	Título	Autor	Data
1094	Network File System Protocol Specification(Especificação do Protocolo de Sistema de Arquivos de Rede)	Sun Microsystems	março de 1989
1813	NFS Version 3 Protocol Specification (Especificação do Protocolo NFS Versão 3)	B. Callaghan et al.	junho de 1995
2623	NFS Version 2 and Version 3 Security Issues (Questões de Segurança nas Versões 2 e 3 do NFS)	M. Eisler	junho de 1999
2624	NFS Version 4 Design Considerations (Considerações de Projeto do NFS Versão 4)	S. Shepler	junho de 1999

17.10 Exercícios

★ **E17.1** Quais são as vantagens de automont? Forneça arquivos de configuração para **automount** e **amd** para a situação apresentada em E17.4.

★ **E17.2** Explore sua configuração NFS local. O NFS é usado ou há uma outra solução em seu lugar? A automontagem é usada? Que decisões foram tomadas?

★ **E17.3** Qual é a relação entre **mountd**, **nfsd** e **portmap**? Qual o significado da dependência do NFS em relação a **portmap** em termos de segurança?

★★ **E17.4** Quais são algumas das implicações de projeto do NFS ser um protocolo sem estados?
Discuta, em particular, qualquer efeito que esta condição de ausência de estados tem sobre o bloqueio de arquivos, permissões de acesso e segurança. Em que um sistema de arquivos de rede com estados diferiria do NFS?

★★ **E17.5** Seu empregador precisa exportar **/usr** e **/usr/local** via NFS.
Dê as seguintes informações.

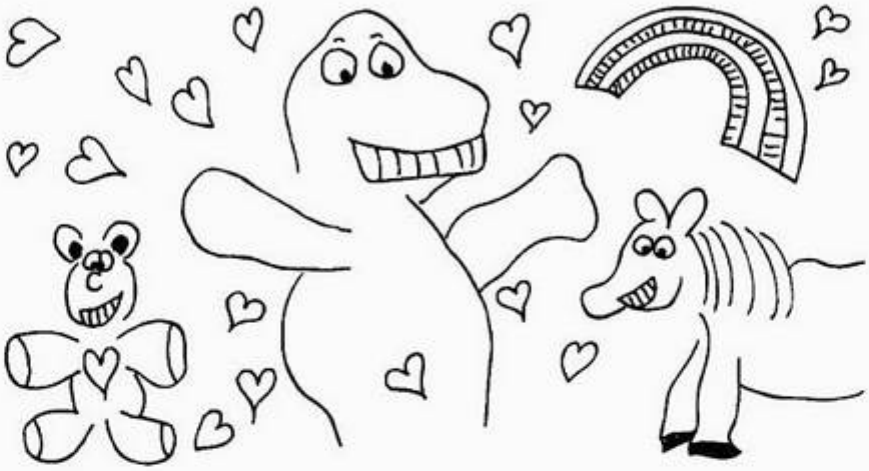
a) Devido à política da empresa, você quer que somente o seu departamento (sub-rede local 192.168.123.0/24) seja capaz de usar estes sistemas de arquivos exportados. Que linhas teriam de ser adicionadas a quais arquivos para implementar esta configuração? Preste atenção às opções de exportação apropriadas.

b) Liste as etapas necessárias para fazer com que **mountd** e **nfsd** reconheçam estes novos sistemas de arquivos compartilhados. Como você poderia verificar se os diretórios estavam sendo compartilhados sem ter de montá-los?

c) Descreva uma estratégia que faria com que todas as máquinas de sua sub-rede local montassem automaticamente os diretórios exportados usando os pontos de montagem **/mnt/usr** e **/mnt/usr/local**.

Snow666

18 Compartilhando Arquivos de Sistema



Um sistema funcionando de maneira adequada depende de dezenas, quiçá centenas, de arquivos de configuração todos contendo os trechos corretos de informação. Ao multiplicarmos o número de arquivos de configuração em um host pelo número de hosts numa rede, o resultado pode se transformar em milhares de arquivos — um número muito grande para ser gerenciado manualmente.

No mundo prático, as máquinas normalmente são similares sob o ponto de vista administrativo. Em vez de editar arquivos de texto em cada máquina, você verá que é mais eficiente combinar máquinas em grupos que compartilhem informações de configuração. Você poderá combinar máquinas de várias formas diferentes.

A maneira mais simples é preservar uma cópia mestre de cada arquivo de configuração em um local e distribuí-la aos membros do grupo sempre que for alterada. Essa solução possui as vantagens de ser simples e de funcionar em qualquer sistema Linux (e UNIX).

Uma outra forma seria eliminar todos os arquivos de texto e fazer com que cada máquina obtivesse suas informações de configuração de um servidor central. Isso é mais complicado do que copiar arquivos, porém soluciona outros problemas também. Por exemplo, os clientes não podem perder atualizações, mesmo se estiverem fora do ar quando acontecer uma modificação. Talvez também seja mais rápido obter informações de um servidor do que de um arquivo, dependendo da velocidade do disco local e do nível de caching executado pelo servidor. Por outro lado, toda a rede pode ficar pendurada quando o servidor central apresentar problemas.

Foram feitas várias tentativas para desenvolver bancos de dados administrativos para redes grandes e todas elas são sistemas interessantes. Porém, nenhum dos produtos atuais parece exatamente correto em sua abordagem. Alguns são simples, mas inseguros e não podem ser expandidos. Outros são funcionais, porém difíceis de serem manejados. Alguns parecem promissores, mas ainda não estão totalmente prontos.

Todos os sistemas parecem ter limitações que podem impedi-lo de configurar a rede da maneira que você deseja e nenhum deles irá gerenciar todas as informações que talvez você queira compartilhar entre suas máquinas.

Neste capítulo discutiremos primeiramente algumas técnicas básicas para manter os arquivos sincronizados dentro de uma rede. A seguir falaremos sobre o sistema de banco de dados mais largamente usado, o NIS, e mencionaremos brevemente seu irmão mais velho, o NIS+ (que não é usado no Linux e está fadado a ser descontinuado). Por fim, trataremos do LDAP, um sistema relativamente novo que pode se tornar amplamente empregado ao longo dos próximos anos. Os grandes sites normalmente acabam usando vários desses métodos e sistemas em combinação.

Snow666

18.1 O Que Deve Ser Compartilhado

Dos diversos arquivos de configuração num sistema Linux, somente um subconjunto deles pode ser compartilhado de forma útil entre as máquinas. Os arquivos mais comumente compartilhados são apresentados na Tabela 18.1.

Tabela 18.1 Arquivos de sistema comumente compartilhados.

Nome do arquivo	Função
/etc/passwd	Banco de dados de informações de contas de usuário
/etc/shadow[a]	Arquivo de senhas-sombra
/etc/group	Definições de grupos UNIX
/etc/hosts	Associações entre nomes de host e endereços IP
/etc/networks	Associa nomes de texto a números de rede IP
/etc/services	Lista números de portas para serviços de rede bem conhecidos
/etc/protocols	Associa nomes de texto a números de protocolo
/etc/ethers	Associação entre nomes de host e endereços Ethernet
/etc/mail/aliases	Contém nomes alternativos de e-mails
/etc/rpc	Lista números de ID para serviços RPC
/etc/netgroup	Define conjuntos de hosts, usuários e redes
/etc/printcap	Banco de dados de informações de impressoras

a. Não necessariamente compartilhável com outras variantes do UNIX, já que o tipo de criptografia pode variar; veja a página 67.

Muitos outros arquivos de configuração poderiam ser compartilhados entre os sistemas, dependendo do nível de similaridade que você deseja ter em seu site. Em grande parte, esses outros arquivos de configuração estão associados a aplicativos específicos (por exemplo, **/etc/sendmail.cf** para **sendmail**) e não são suportados por sistemas de bancos de dados administrativos como o NIS e o LDAP; você terá de compartilhar os arquivos através de suas cópias.

Consulte a seção 21.4 para obter maiores informações sobre o PAM.

Historicamente, muitos dos arquivos apresentados na Tabela 18.1 têm sido acessados através de rotinas na biblioteca C padrão. Por exemplo, o arquivo **/etc/passwd** pode ser pesquisado com as rotinas **getpwuid**, **getpwnam** e **getpwent**. Essas rotinas cuidam da abertura, leitura e análise sintática do arquivo **passwd** para que os programas em nível de usuário não tenham de fazer isto por conta própria. As distribuições modernas do Linux também usam PAM (Pluggable Authentication Modules, módulos de autenticação plugáveis), que fornecem uma interface de programação padrão para realizar pesquisas relacionadas com segurança. Os módulos PAM permitem que sistemas de autenticação como o Kerberos sejam facilmente integrados no Linux.

Sistemas de bancos de dados administrativos complicam as coisas fornecendo fontes alternativas para grande parte destas informações. As rotinas de biblioteca C tradicionais (**getpwent** etc.) estão cientes dos sistemas de bancos de dados comuns e podem acessá-los além dos (ou em vez dos) arquivos planos padrão. O complemento exato de fontes de dados que são consultadas é configurado pelo administrador do sistema; veja o tópico *Priorizando fontes de informações administrativas* na seção 18.3 para maiores detalhes.

nscd: coloque em cache os resultados das pesquisas

Em algumas distribuições (entre nossos exemplos, somente o SuSE), mais um intruso na festa de arquivos de sistema é o **nscd**, intitulado de maneira um tanto errônea de daemon de cache de serviço de nomes. O **nscd** funciona em conjunto com a biblioteca C para colocar em cache os resultados de chamadas de biblioteca como **getpwent**. **nscd** é simplesmente um envoltório para essas rotinas de biblioteca; ele não sabe nada a respeito das fontes de dados que estão sendo efetivamente consultadas. **nscd** deveria, teoricamente, melhorar o desempenho das pesquisas, mas em geral ele não leva a uma melhoria subjetiva tão grande sob o ponto de vista do usuário.

Veja o Capítulo 16 para obter maiores informações sobre o DNS.

O "daemon de cache de serviço de nomes" é potencialmente enganoso, pois o termo "serviço de nomes" normalmente se refere ao DNS, o sistema de banco de dados distribuído que faz a associação entre nomes de hosts e endereços Internet. **nscd** efetivamente coloca em cache os resultados das pesquisas DNS (pois envolve **gethostbyname** etc.), mas também envolve as rotinas de bibliotecas que acessam informações dos arquivos **passwd** e **group** e seus equivalentes de bancos de dados de rede. (Por questões de segurança, as pesquisas em **/etc/shadow** não são colocadas em cache.)

www.snow666.blogspot.com
Snow666

Teoricamente, **nscd** não deve apresentar nenhum efeito na operação do sistema a não ser acelerar pesquisas repetidas. Na prática, ele pode provocar comportamentos inesperados, pois mantém sua própria cópia de resultados da pesquisa. As pesquisas são armazenadas no cache por um período fixo de tempo (definida no arquivo de configuração do **nscd**, **/etc/nscd.conf**) e sempre há a possibilidade de que alterações recentes não se reflitam no cache do **nscd** até que os dados anteriores tenham se expirado. **nscd** é inteligente o bastante para monitorar fontes de dados locais (como **/etc/passwd**) em busca de alterações, de modo que atualizações locais devem ser propagadas dentro de 15 segundos. Para entradas remoas, como aquelas recuperadas através de NIS, talvez seja necessário aguardar o período de timeout completo antes de as alterações entrarem em efeito.

Em sistemas que o utilizam, o **nscd** inicia no momento da inicialização do sistema e é executado continuamente. O padrão **/etc/nscd.conf** especifica um timeout de 10 minutos para dados **passwd** e de uma hora para **hosts** e **group**, com um timeout negativo de 20 segundos (o período de tempo antes de uma pesquisa malsucedida ser novamente executada). Na prática, esses valores raramente precisam ser alterados. Caso se veja na situação de ficar tentando adivinhar por que uma alteração feita recentemente não é mostrada, provavelmente o motivo seja o **nscd**.

18.2 Copiando Arquivos

Nós usamos a cópia de arquivos de "força bruta" para manter a rede de Engenharia da Universidade do Colorado. Ela não é uma solução elegante, porém funciona em qualquer tipo de máquina e é fácil de ser configurada e mantida. Também é um sistema confiável, pois minimiza as dependências entre as máquinas (embora ele também facilite que máquinas saiam de sincronismo). A cópia de arquivos oferece ainda a máxima flexibilidade em termos do que pode ser distribuído e como. De fato, ela também é usada freqüentemente para manter arquivos de dados e aplicativos atualizados, bem como arquivos de sistema.

Bem poucos arquivos de configuração não são suportados por qualquer um dos serviços de bancos de dados comuns. (Entre os exemplos, podemos citar **/etc/sendmail.cf**, que determina como **sendmail** entrega e-mails e **/etc/ntp.conf**, que determina como os hosts mantêm seus relógios sincronizados.) Caso queira manter esses arquivos em sincronismo (o que normalmente é uma boa postura), você realmente não tem nenhuma escolha a não ser usar algum tipo de sistema de cópia por força bruta mesmo que você distribua outras informações com NIS ou LDAP.

Normalmente é suposto em manuais e na cultura do Linux que você usará um sistema como o NIS caso um esteja disponível. Entretanto, se suas necessidades não forem complexas, você não precisará de uma solução complexa. Algumas vezes a solução mais simplória e imediata é a melhor. Nosso site é formado por vários feudos conectados, porém independentes. Somente uma pequena parcela de dados administrativos é compartilhada em todos eles. Em nosso esquema de distribuição, cada feudo possui um ou dois servidores que armazenam cópias mestras de seus arquivos de sistema. Esse é o tipo de ambiente no qual a cópia de arquivos funciona bem, já que a tarefa é simplesmente bombear dados, não ajustá-los para máquinas ou redes específicas.

Os sistemas de cópia de arquivos podem usar tanto um modelo "empurrão" quanto um modelo "puxão". Com empurrão, o servidor mestre distribui periodicamente os arquivos mais recentes a cada cliente, independentemente do cliente querê-los ou não. Os arquivos podem ser empurrados explicitamente sempre que uma modificação for feita ou poderão ser simplesmente distribuídos segundo um cronograma regular (talvez com alguns arquivos sendo transferidos com maior freqüência do que outros).

O modelo empurrão possui a vantagem de manter o sistema de distribuição centralizado numa máquina. Arquivos, listas de clientes, scripts de atualização e cronogramas são mantidos num único local, facilitando o controle. Uma desvantagem é que cada cliente tem de permitir que o mestre modifique seus arquivos de sistema, criando um risco de segurança.

Num sistema puxão, cada cliente é responsável pela própria atualização no servidor. Essa é uma forma menos centralizada de distribuição de arquivos, porém ela também é mais adaptável e segura. Um sistema puxão é especialmente interessante quando os dados são compartilhados através de fronteiras administrativas, pois as máquinas mestras e cliente não precisam ser executadas pela mesma facção política.

rdist: distribuição de arquivos

O comando **rdist** é a maneira mais fácil de distribuir arquivos de um servidor central. Ele possui certa semelhança com **make**: é usado um editor de texto para criar uma especificação dos arquivos a serem distribuídos e depois se usa **rdist** para alinhar a realidade com a especificação feita. **rdist** copia arquivos somente quando estão desatualizados, para que possamos criar uma especificação como se todos os arquivos devessem ser copiados e deixar que **rdist** otimize o trabalho desnecessário.

Snow666

rdist preserva informações relativas ao proprietário, grupo, modo e horário de modificação dos arquivos. Quando **rdist** atualiza um arquivo existente, ele primeiramente elimina a versão antiga antes de instalar a nova. Esse recurso faz com que **rdist** seja adequado para transferência de executáveis que poderiam estar sendo usados durante a atualização.[1]

Infelizmente, **rdist** historicamente tem apresentado alguns problemas relativos à segurança. Ele era tradicionalmente executado sobre o **rsh** e usava autenticação tipo **rsh** para ganhar acesso a sistemas remotos. Nesse esquema, o acesso root é permitido em qualquer host listado no arquivo **/.rhosts** do sistema desejado. Esse esquema era aceitável nas redes relativamente abertas e isoladas do passado, porém é muito perigoso no contexto da Internet. Se um host for invadido, a segurança de todos os hosts que confiam nele estará automaticamente comprometida.

Já que estamos distribuindo arquivos administrativos como **/etc/passwd**, é quase certo de que o acesso root no servidor mestre poderá ser explorado para acesso root nos clientes. Esse não é o problema. O problema é que ao se executar **rlogind** (o servidor para **rsh** e também **rlogin** e **rcp**), máquinas-cliente acabam ficando vulneráveis a outros tipos de ataque.

Versões atuais de **rdist** são melhores pelo fato de permitirem que qualquer outro comando que compreenda a mesma sintaxe possa ser um substituto do **rsh**. Na prática, o substituto é **ssh**, que usa criptografia para verificar a identidade dos hosts e para impedir escutas na rede de obterem cópias de seus dados. O lado negativo é que você tem de rodar servidores **ssh** remotos num modo que não exige o emprego de senhas. Trata-se de uma configuração menos segura do que normalmente recomendaríamos, porém ainda é um imenso progresso em relação a **rsh**. Consulte a página 502 para obter maiores informações sobre **sshd** e seus modos de autenticação.

Agora que já esgotamos os perigos de **rdist**, vamos ver como ele funciona efetivamente. Como **make**, **rdist** procura um arquivo de controle (**Distfile** ou **distfile**) no diretório atual. **rdist -f** *distfile* especifica explicitamente o nome do caminho do arquivo de controle. Dentro de **Distfile**, são usados intercambiavelmente como separadores tabulações, espaços e linhas novas. Os comentários são introduzidos por um símbolo #.

O miolo de um **Distfile** consiste em instruções na forma

```
rótulo: nomes_de_caminhos -> destinos comandos
```

O campo *rótulo* associa um nome à instrução. Por meio do shell, é possível executarmos **rdist** *rótulo* para distribuir somente os arquivos descritos numa dada instrução.

nomes_de_caminhos e *destinos* são, respectivamente, listas de arquivos a serem copiados e hosts para os quais copiar. Se houver mais do que uma entrada numa lista, ela pode ser envolvida por parênteses e os elementos têm de ser separados por espaços em branco. *nomes_de_caminhos* pode incluir caracteres de substituição estilo shell (por exemplo, **/usr/man/man[123]** ou **/usr/lib/***). A notação *~usuario* também é aceitável, porém ela é avaliada separadamente nas máquinas de origem e de destino.

Como padrão, **rdist** copia os arquivos e diretórios listados em *nomes_de_caminho* para os caminhos equivalentes em cada máquina de destino. Podemos modificar esse comportamento fornecendo uma seqüência de comandos e terminando cada um deles com um ponto-e-vírgula.

São compreendidos os seguintes comandos:

```
install opções [diretório_destino];
notify lista_de_nomes;
except lista_de_caminhos;
except_pat lista_de_padrões;
special [lista_de_caminhos] string;
```

O comando `install` configura opções que afetam a maneira pela qual **rdist** copia arquivos. As opções controlam tipicamente o tratamento de links simbólicos, a correção do algoritmo de verificação de diferenças de **rdist** e a maneira pela qual as eliminações são manipuladas. As opções deveriam ser precedidas por `-o` e são formadas por uma lista de nomes de opções separadas por vírgulas. Por exemplo, a linha

```
install -oremove,follow ;
```

faz com que **rdist** siga links simbólicos (em vez de simplesmente copiá-los como links) e elimina arquivos existentes da máquina de destino que não possuem nenhum equivalente na máquina de origem. Veja a página de manual para **rdist** para obter uma lista completa das opções. Os padrões são quase sempre aquilo que desejamos.

---

1. Embora a versão antiga desapareça do espaço de nomes do sistema de arquivos, ela continua a existir até que todas as referências tenham sido liberadas. Você também tem de estar ciente desse efeito ao gerenciar arquivos de log. Veja a seção 11.1, para obter maiores informações.

www.sn0w666.blogspot.com
Sn0w666

O nome "install" é um tanto impreciso, já que os arquivos são copiados independentemente de um comando `install` estar presente ou não. São especificadas opções como seriam na linha de comandos **rdist**, porém quando inclusas em **Distfile**, elas se aplicam somente ao conjunto de arquivos manipulados por este comando `install`.

O *diretório_destino* opcional especifica um diretório de instalação nos hosts de destino. Como padrão, **rdist** usa os nomes de arquivos originais.

O comando `notify` usa uma lista de endereços de e-mail como seu argumento. **rdist** envia e-mails a esses endereços sempre que um arquivo é atualizado. Qualquer endereço que não contiver um símbolo de arroba (@) receberá como sufixo o nome do host de destino. Por exemplo, **rdist** expandiria "pete" para "pete@anchor" ao relatar uma lista de arquivos atualizada no host anchor.

Os comandos `except` e `except_pat` eliminam nomes de caminhos da lista de arquivos a serem copiados. Argumentos para `except` devem ser literalmente coincidentes, e aqueles para `except_pat` são interpretados como expressões regulares. Esses comandos de exceção são úteis, pois **rdist**, assim como **make**, permite a definição de macros no início de seu arquivo de controle. Talvez você queira usar uma lista similar de arquivos para várias instruções, especificando somente os acréscimos e eliminações para cada host.

O comando `special` executa um comando **sh** (o argumento *string*, que deve ser colocado entre aspas) em cada host remoto. O comando é executado após cada arquivo ser copiado. Infelizmente não há nenhuma maneira de executar um comando somente depois de todos os arquivos terem sido copiados.

Eis um exemplo simples de um **Distfile**:

```
SYS_FILES = (/etc/passwd /etc/group /etc/mail/aliases)
GET_ALL   = (chimchim lollipop barkadon)
GET_SOME  = (whammo spiff)

all: ${SYS_FILES} -> ${GET_ALL}
   notify barb;
   special /etc/mail/aliases "/usr/bin/newaliases";

some: ${SYS_FILES} -> ${GET_SOME}
   except /etc/mail/aliases;
   notify eddie@spiff;
```

*Consulte a seção 19.4 para mais informações sobre **newaliases**.*

Essa configuração replica os três arquivos de sistema listados em chimchim, lollipop e barkadon e envia um e-mail para barb@*destino* descrevendo quaisquer atualizações ou erros que venham a ocorrer. Depois de **/etc/mail/aliases** ser copiado, **rdist** executa **newaliases** em cada um dos destinos. Somente dois arquivos são copiados para whammo e spiff. **newaliases** não é executado, e um relatório é enviado via e-mail para eddie@spiff.

Para que **rdist** funcione entre as máquinas, você também tem de fazer com que **sshd** nos hosts receptores confiem no host do qual você está distribuindo os arquivos. Para tanto, geramos uma chave em puro texto para o host mestre e armazenamos uma cópia da porção pública no arquivo **~root/.ssh/authorized_keys** em cada receptor. Provavelmente talvez também seja interessante restringir o que esta chave é capaz de fazer e onde ela pode fazer o login. Veja a descrição do "método B" na seção 21.9, para obter maiores informações.

rsync pode ser encontrado em rsync.samba.org.

rsync: transferência de arquivos de maneira mais segura

rsync, criado por Andrew Tridgell e Paul Mackerras, é similar em espírito a **rdist**, porém com um foco um tanto diferente. Ele não usa um arquivo de controle de cópia de arquivos da maneira como faz **rdist** (embora no servidor realmente haja um arquivo de configuração). **rsync** é uma espécie de versão envenenada de **rcp** que é escrupulosa em relação a preservar links, horários de modificação e permissões. É mais eficiente em termos de rede do que **rdist**, pois ele procura dentro de arquivos individuais e tenta transmitir somente as diferenças entre versões. A maioria das distribuições do Linux fornece **rsync**, inclusive Red Hat, SuSE e Debian.

Sob nossa perspectiva, a principal vantagem de **rsync** é o fato de máquinas receptoras poderem fazer o lado remoto como um processo de servidor proveniente de **inetd**. O servidor (na verdade apenas um modo diferente de **rsync**, que obrigatoriamente deve ser instalado tanto no mestre quanto nos clientes) é bastante configurável: ele pode restringir o acesso remoto a um determinado conjunto de diretórios e pode exigir que o mestre prove sua identidade através de uma senha. Já que não é necessário nenhum acesso **rsh**, podemos configurar **rsync** para distribuir arquivos de sistema sem muito comprometimento da segurança. (Entretanto, caso prefira usar **rsh** ou **ssh** em vez de um processo de servidor baseado em **inetd**, **rsync** permite que você faça isto também.) Além disso, **rsync** também

www.snow666.blogspot.com
Snow666

pode ser executado no modo puxão (puxando arquivos do servidor **rsync**, em vez de empurrá-los para ele), que é até mais seguro (veja a seção *Puxando arquivos*, mais à frente).

Infelizmente, **rsync** não é tão flexível quanto **rdist** e sua configuração é menos sofisticada do que o **distfile** de **rdist**. Não podemos executar comandos arbitrários nos clientes e não podemos aplicar **rsync** a vários hosts de uma vez.

Por exemplo, o comando

```
# rsync -gopt --password-file=/etc/rsync.pwd /etc/passwd lollipop::sysfiles
```

transfere o arquivo **/etc/passwd** para a máquina lollipop. As opções **-gopt** preservam as permissões, propriedades e os horários de modificação do arquivo. Os dois-pontos duplicados em **lollipop::sysfiles** fazem com que **rsync** contacte o **rsync** remoto diretamente na porta 873 em vez de usar **rsh**. A senha armazenada em **/etc/rsync.pwd** autentica a conexão.[2]

Esse exemplo transfere somente um arquivo, porém **rsync** é capaz de manipular vários arquivos de uma só vez. Além disso, os flags **--include** e **--exclude** permitem que especifiquemos uma lista de expressões regulares para fazer a concordância com nomes de arquivos, de modo que podemos configurar um conjunto de critérios de transferência relativamente sofisticado. Se a linha de comando se tornar muito extensa, você também poderá ler os padrões de arquivos separados através das opções **--include-file** e **--exclude-file**.

O Red Hat vem com o **rsync** pronto para ser usado, porém desativado. Para ativá-lo, edite o arquivo **/etc/xinetd.d/rsync** e altere a linha que diz `disable=yes` para `disable=no`.

O SuSE inclui uma linha para **rsyncd** em **/etc/inetd.conf**, porém ela está comentada por padrão. Elimine esse comentário (elimine o símbolo # que precede a linha) para habilitar **rsyncd**.

O pacote **rsync** do Debian não configura o servidor **rsync** como padrão; você tem de criar um arquivo **rsyncd.conf** apropriado e acrescentar uma entrada a **/etc/inetd.conf**. Por exemplo, a linha a seguir em **inetd.conf** acionará o servidor (com envoltórios TCP):

```
rsync stream tcp nowait root /usr/sbin/tcpd /usr/bin/rsyncd --daemon
```

Consulte a seção 21.8 para obter maiores informações sobre envoltórios TCP.

Se você usa envoltórios TCP (TCP wrappers), talvez queira configurar **tcpd** para bloquear o acesso de todos os hosts, exceto aquele que estará distribuindo os arquivos de sistema. A rejeição de hosts também pode ser especificada em **rsyncd.conf**, porém não custa nada erguer várias barreiras.

Assim que tiver habilitado **rsync**, você precisará configurar uma série de arquivos de configuração para informar o servidor **rsync** como se comportar. O principal arquivo é **/etc/rsyncd.conf**, que contém tanto parâmetros de configuração globais como um conjunto de "módulos", cada um dos quais é uma árvore de diretórios para exportação ou importação. Uma configuração razoável para um módulo que se pode empurrar (isto é, que aceitará transferências de arquivos que chegam iniciadas pelo cliente com que se estabelece a conexão) será parecida com o seguinte:

```
# sysfiles é apenas um título arbitrário para o módulo em particular.
[sysfiles]
# Este é o caminho para o qual você dá permissão que os arquivos sejam encaminhados. Poderia ser apenas /.
path = /etc
# Este é o arquivo de segredos contendo o par nome de usuário/senha para autenticar o módulo
secrets file = /etc/rsyncd.secrets
# Pode ser lido somente se você estiver puxando arquivos
read only = false
# UID e GID sob os quais a transferência será feita
uid = root
gid = root
# Lista de hosts que têm a permissão para se conectar
hosts allow = nome_do_host_mestre_de_distribuição
```

Muitas outras opções podem ser configuradas, porém os padrões são razoáveis. Essa configuração limita as operações ao diretório **/etc** e permite o acesso somente pelo host listado. Sob a perspectiva do usuário ou do cliente, podemos enviar arquivos via **rsync** para o servidor com o destino *nome_do_host*::**sysfiles**, que se associa ao módulo

2. Embora a senha não seja enviada em texto puro pela rede, os arquivos transferidos não são criptografados. Se você usar **ssh** como transporte (**rsync -gopt -e ssh /etc/passwd /etc/shadow lollipop:/etc** — observe a ocorrência única dos dois-pontos), a conexão será criptografada, embora **sshd** terá de ser configurado para não exigir uma senha. Escolha a sua bebida!

Snow666

anterior. Se você quiser configurar **rsync** no modo puxão (puxando arquivos de um servidor **rsync** central), a configuração acima ainda funcionaria, muito embora talvez você queira restringir um pouco mais as coisas (por exemplo, configurando o modo de transferência para somente leitura).

A última coisa necessária é configurar um arquivo **rsyncd.secrets**. Ele geralmente é mantido em **/etc** (embora possamos colocá-lo em outro local) e contém as senhas que os clientes poderão usar para se autenticar. Por exemplo:

```
root:senha
```

As senhas **rsync** devem, como regra geral, ser diferentes das senhas normais. Pelo fato de serem mostradas no formato texto puro, **rsyncd.secrets** deverá ser legível somente pelo root.

Puxando arquivos

Há várias maneiras de implementar um sistema puxador. A mais fácil e objetiva é fazer com que os arquivos fiquem disponíveis numa central FTP ou servidor Web e fazer com que os clientes automaticamente realizem o download dos arquivos conforme a necessidade. Em tempos remotos, os administradores nomeariam os seus próprios utilitários para fazerem isso (normalmente criando scripts para **ftp** com um sistema como **expect**), porém agora há utilitários-padrão que realizam isto para você.

Um utilitário popular que vem com a maioria das distribuições Linux é o **wget**. Ele é um pequeno programa objetivo que pesquisa o conteúdo de um URL (seja FTP ou HTTP). Por exemplo, para transferir um arquivo via FTP com **wget**, basta executar

```
wget ftp://usuário:senha@nome_do_host/caminho/para/arquivo
```

O arquivo (*arquivo*) especificado será depositado no diretório atual.

Uma outra alternativa somente para FTP é **ncftp**, que também acompanha a maioria das distribuições. Ele é simplesmente um cliente FTP otimizado que permite a utilização de scripts de modo fácil.

Alguns sites distribuem arquivos publicando-os num sistema de arquivos em rede como o NFS. Esta talvez seja a técnica mais simples sob o ponto de vista de automação — tudo o que é necessário é **cp** (copiar), pelo menos teoricamente. Na prática, provavelmente você vai querer ser um pouco mais sofisticado e verificar sintomas de problemas de segurança e conteúdo corrompido antes de copiar cegamente os arquivos de sistema. Divulgar arquivos de sistema críticos via NFS possui algumas desvantagens sob o ponto de vista de segurança, porém é uma maneira simples e eficiente de transferir informações.

Uma opção final é usar **rsync**. Se rodarmos um servidor **rsync** num host de distribuição central, os clientes poderão simplesmente baixar os arquivos via **rsync**. Esse método talvez seja ligeiramente mais complexo do que usar FTP, porém nos dá acesso a todos os recursos de **rsync**.

Independentemente do sistema usado, tome cuidado para não sobrecarregar o seu servidor de dados. Se um monte de máquinas da rede tentar acessar o servidor ao mesmo tempo (por exemplo, se todo o mundo tentar atualizar **cron** simultaneamente), você poderá provocar inadvertidamente um ataque de recusa de serviços. Sites grandes devem ter esse problema em mente e fornecer escalonamento de tempo ou randomização. Uma maneira simples de fazer isso é envolver tarefas **cron** num script Perl como o indicado a seguir:

```
#!/usr/bin/perl
sleep rand() * 600; # adormecer entre 0 e 600 segundos (isto é, 10 minutos)
system(copy_files_down);
```

18.3 NIS: Network Information Service

O NIS, lançado pela Sun nos anos 80, foi o primeiro banco de dados administrativo de "horário nobre". Ele foi chamado originalmente de Sun Yellow Pages, mas teve de receber um outro nome devido a questões jurídicas. Os comandos do NIS ainda começam com as letras **yp**, portanto, é difícil se esquecer do nome original. O NIS foi largamente adotado entre os fornecedores de UNIX e é suportado por praticamente qualquer distribuição Linux.

A unidade de compartilhamento no NIS é o registro, não o arquivo. Um registro normalmente corresponde a uma linha num arquivo de configuração. Um servidor mestre mantém as cópias autorizadas dos arquivos de sistema, que são mantidas em suas localizações e formatos originais e são editadas com um editor de texto exatamente como antes. Um processo de servidor disponibiliza o conteúdo dos arquivos através da rede. Um servidor e seus clientes formam um "domínio"[3] NIS.

3. Não confunda domínios NIS com domínio DNS. Eles são completamente distinto e não têm nada a ver um com o outro.

www.snow666.blogspot.com
Snow666

Os arquivos de dados são preprocessados nos arquivos de bancos de dados pela biblioteca hashing **gdbm** para melhorar a eficiência das pesquisas. Após editarmos os arquivos no servidor mestre, solicitamos ao NIS para convertê-los a seus formatos hashed usando **make**.

Somente uma chave pode ser associada a cada entrada, de modo que um arquivo de sistema talvez tenha de ser convertido em vários "mapas" NIS. Por exemplo, o arquivo **/etc/passwd** é convertido em dois mapas diferentes chamados **passwd.byname** e **passwd.byuid**. Um mapa é usado para pesquisar entradas pelo nome do usuário e o outro para pesquisar entradas pelo UID. Qualquer um deles pode ser utilizado para enumerar as entradas no arquivo **passwd**. Entretanto, pelo fato de as bibliotecas hashing não preservarem a ordem dos registros, não há nenhuma maneira de reconstruir uma cópia exata do arquivo original (a menos que ele tenha sido ordenado).

O NIS permite que repliquemos os mapas de rede num conjunto de servidores escravos. Disponibilizar mais de um servidor ajuda a aliviar a carga sobre o mestre e a manter os clientes funcionando mesmo quando alguns servidores estiverem fora do ar. Toda vez que um arquivo é alterado no servidor mestre, o mapa NIS correspondente tem de ser enviado para os escravos para que todos os servidores forneçam os mesmos dados. Os clientes não fazem distinção entre servidor mestre e escravo.

Na implementação original do NIS, tínhamos de colocar pelo menos um servidor NIS em cada rede física. Os clientes usam difusão ampla de IP para localizar servidores, e pacotes de difusão ampla não são encaminhados por roteadores e gateways. Entretanto, o comando **ypset** pode ser usado para direcionar um cliente a um determinado servidor; ao primeiro sinal de problemas, o cliente tentará localizar um novo servidor usando difusão ampla. A menos que haja um servidor na rede do cliente, essa seqüência de eventos poderá fazer com que o cliente fique inativo.

Esse sistema provoca uma série de problemas, e dos não menos importante é o fato de ser extremamente inseguro. Um invasor pode configurar um servidor NIS trapaceiro que responda a difusões amplas e forneça dados falsos ou então emita um ataque de recusa de serviços permitindo conexões e depois bloqueando no atendimento das solicitações efetivas. Nos dias de hoje, a técnica de gerenciamento preferida é a de fornecer a cada cliente uma lista explícita de seus servidores NIS legítimos. Esse sistema também possui a vantagem de que os servidores não precisam estar numa sub-rede local. No Linux, os servidores são listados em **/etc/yp.conf**. Eis um exemplo:

```
domain  we.luv.nis  server  10.2.2.3
domain  we.luv.nis  server  10.2.2.4
```

Há uma linha para cada servidor; se um deles falhar, o NIS passará para outro. Note que os servidores são dados na forma de endereços IP. **yp.conf** aceita nomes de hosts, porém estes nomes de hosts têm de ser determinados no momento da inicialização (por exemplo, enumerados no arquivo **/etc/hosts**).

Se você for *obrigado* a usar o modo de difusão ampla, a sintaxe é

```
domain we.luv.nis broadcast
```

Grupos de rede

O NIS introduziu uma abstração popular conhecida como grupos de rede. Os grupos de rede nomeiam conjuntos de usuários, máquinas e redes para fácil referência em outros arquivos de sistema. Eles são definidos em **/etc/netgroup** e também são compartilhados na forma de um mapa NIS.

O formato de uma entrada **netgroup** é

```
nome_do_grupo lista_de_membros
```

Os membros são separados por um espaço em branco. Um membro pode ser um nome de grupo de rede ou uma trinca na forma

```
(nome_do_host,nome_do_usuário,nome_do_domínio)
```

Qualquer campo vazio numa trinca é um curinga; portanto, a entrada `(boulder,,)` se refere a todos os usuários em todos os domínios do host boulder (ou para o próprio host boulder, dependendo do contexto no qual o grupo de rede é usado). Um traço num campo indica negação, ou seja, a entrada `(boulder,-,)` se refere à máquina boulder e nenhum usuário. As definições de grupos de rede podem ser aninhadas.

Veja um exemplo simples de um arquivo **/etc/netgroup**:

```
bobcats         (snake,,) (headrest,,)
servers         (anchor,,) (moet,,) (piper,,) (kirk,,)
anchorclients   (xx,,) (watneys,,) (molson,,)
beers           (anchor,,) (anchor-gateway,,) anchorclients
allhosts        beers bobcats servers
```

Snow666

Esses grupos de rede são definidos em termos de hosts; isto é típico para uso no mundo real.

Veja o Capítulo 17 para obter maiores informações sobre o NFS.

Os grupos de rede podem ser usados em vários arquivos de sistema que definem permissões. A aplicação mais comum hoje é na configuração de exportações NFS. Os grupos de rede podem ser mencionados no arquivo **/etc/exports** para especificar grupos de hosts que têm permissão para montar cada sistema de arquivos. Esse recurso é muito prático quando estamos exportando para um grande número de hosts, particularmente em sistemas que requerem nomes de domínios totalmente qualificados e que limitam as linhas no arquivo **exports** a 1.024 caracteres.

Os grupos de rede são uma boa idéia. Eles simplificam os arquivos de sistema, tornando-os mais compreensíveis. Eles também acrescentam uma camada dissimulada que permite que o estado de um usuário ou máquina seja modificado em um 1 em vez de em 15.

Priorizando fontes de informações administrativas

As informações de configuração podem ser distribuídas de diversas formas. Cada sistema compreende arquivos planos e sabe como usar o DNS para pesquisar nomes de hosts e endereços Internet. A maioria também compreende o NIS. Já que pode haver várias fontes potenciais para um dado bloco de informações, o Linux fornece uma maneira para especificarmos as fontes que devem ser verificadas e a ordem na qual estas verificações devem ser feitas.

Na implementação original do NIS (anterior ao Linux), alguns arquivos de configuração (particularmente os arquivos **/etc/passwd** e **/etc/group**) tinham de ser configurados para "convidar" os conteúdos dos mapas NIS correspondentes. O convite era estendido por inclusão de encantos especiais dentro dos próprios arquivos. Um sinal "+" isolado no início de uma linha incluiria todo o mapa NIS, "+@*grupo_de_rede*" incluiria somente as entradas relevantes a um dado grupo de rede e "+*nome*" incluiria uma única entrada.

Essa abordagem jamais foi benquista e foi suplantada por um arquivo de configuração, **/etc/nsswitch.conf**, que permite a especificação de um caminho de busca explícito para cada tipo de informação administrativa. O comportamento original pode ser emulado através do uso do modo de compatibilidade, mas é pouco provável que você realmente queira alguma vez fazer tal emulação. (Infelizmente, a emulação é o padrão na maioria das distribuições.)

Um arquivo **nsswitch.conf** típico seria parecido com o seguinte:

```
passwd:   files nis
hosts:    files dns
group:    files
```

Cada linha configura um tipo de informação (normalmente, um equivalente ao arquivo plano). As fontes potenciais são `nis`, `nisplus`, `ldap`, `hesiod`, `files`, `dns` e `compat`; elas se referem, respectivamente, ao NIS, NIS+, LDAP, serviço de diretórios Hesiod do MIT, arquivos planos comuns (ignorando identificadores como "+"), DNS, arquivos planos com "convite" NIS (honrando "+"). O DNS é uma fonte de dados válida somente para informações de host e de rede. Uma outra fonte (infelizmente não bem documentada) é `db`, que lê a versão hashed (**gdbm**) do mapa de **/var/db** (por exemplo, **/var/db/passwd.db**). Se os arquivos planos forem grandes, o emprego de versões hashed poderia aumentar substancialmente a velocidade de pesquisa.

As fontes são tentadas da esquerda para a direita até que uma delas produza uma resposta para a consulta. No exemplo anterior, a rotina **gethostbyname** verificaria primeiramente o arquivo **/etc/hosts** e se o host não estivesse listado lá, ela iria então verificar o DNS. Por outro lado, consultas sobre grupos UNIX, verificariam somente o arquivo **/etc/group**.

Se necessário, podemos definir a "falha" de uma fonte mais especificamente colocando expressões entre colchetes após a falha. Por exemplo, a linha

```
hosts: dns [NOTFOUND=return] nis
```

faz com que seja usado DNS exclusivamente se ele estiver disponível; uma resposta negativa do servidor de nomes faz com que as consultas retornem imediatamente (com um código de falha) sem verificar o NIS. Entretanto, o NIS será usado se não estiver disponível nenhum servidor de nomes. Os vários tipos de falhas são mostrados na Tabela 18.2; cada uma delas pode ser configurada para `return` ou `continue`, indicando se a consulta deve ser abortada ou encaminhada para a próxima fonte.

Snow666

Tabela 18.2 Modos de falha reconhecidos em /etc/nsswitch.conf.

Condição	Significado
UNAVAIL	A fonte não existe ou está fora do ar.
NOTFOUND	A fonte existe, porém não pode responder à consulta.
TRYAGAIN	A fonte existe porém está ocupada.
SUCCESS	A fonte foi capaz de responder à consulta.

Como padrão, todas as distribuições Linux vêm com arquivos **nsswitch.conf** que são razoáveis para uma máquina dedicada sem NIS. Todas as entradas vão para arquivos planos, exceto as pesquisas de host, que primeiramente consultam arquivos planos e depois o DNS. As distribuições usadas como exemplo neste livro apresentam como padrão o modo `compat` para **passwd** e **group**, que provavelmente vale a pena ser mudado. Se você realmente usa NIS, coloque-o apenas no arquivo **nsswitch.conf**.

O Debian vem com `protocols`, `services`, `ethers` e `rpc` indo para `db` e depois `files`. Isso é um pouco estranho, já que o Debian não inclui, de fato, **/var/db** ou qualquer outro mecanismo para mantê-lo. Provavelmente seria um pouco mais eficiente ir diretamente para `files`; você pode modificar os ajustes de configuração para fazer isto caso queira.

Vantagens e desvantagens do NIS

Um excelente recurso do NIS é que ele pode ser compreendido por meros mortais. O NIS é análogo a copiar arquivos; na maioria dos casos, não é necessário que os administradores estejam cientes dos formatos de dados internos do NIS. A administração é realizada com os mesmos velhos arquivos planos e somente um ou dois novos procedimentos precisam ser aprendidos.

Já que não há nenhum modo de fazer o link de domínios NIS, o NIS não é adequado para gerenciamento de uma grande rede de máquinas a menos que uma única configuração seja aplicada a cada máquina. Podemos dividir uma rede grande em vários domínios NIS, porém cada domínio tem de ser administrado separadamente. Mesmo se uma rede grande efetivamente usar uma única configuração, limitações na expansão de servidores escravos significam que na prática estes sites normalmente terão algum outro tipo de mecanismo para manter seus servidores NIS em sincronia. Eles normalmente acabam rodando seus próprios bancos de dados auxiliares e fazendo com que seus servidores NIS atualizem seus dados com base nesta fonte central.

*Veja o Capítulo 9 para obter maiores informações sobre **cron**.*

Se um servidor escravo estiver fora do ar ou inacessível quando um mapa for alterado, a cópia do escravo não será atualizada. Os escravos têm de sondar o mestre periodicamente para se certificarem de que eles possuam a versão mais recente de cada mapa. Embora sejam fornecidas com o NIS ferramentas básicas para sondagem, você tem de implementar o esquema desejado usando **cron**. Mesmo assim, há a possibilidade de que duas versões diferentes de um mapa sejam servidas simultaneamente por um instante, com os clientes aleatoriamente vendo uma ou outra.

O NIS não é seguro. O modo de difusão ampla é particularmente ruim; qualquer host numa rede pode se dizer servidor de um determinado domínio e alimentar dados administrativos adulterados aos clientes NIS. Podemos evitar esse problema enumerando explicitamente os servidores NIS permissíveis para cada cliente.

Podemos restringir os hosts capazes de ler os mapas de um servidor listando-os explicitamente no arquivo **/var/yp/securenets**; entretanto, isso não é 100% seguro. Também podemos aumentar a segurança de um sistema distribuindo o arquivo de senhas-sombra com alguma outra técnica (como **rdist** ou **rsync**); não recomendamos o emprego de NIS para servir senhas-sombra.

Versões mais antigas do NIS do Linux (inclusive aquelas contidas no pré-7.* SuSE e Red Hat) contêm brechas de segurança conhecidas. Caso esteja rodando um sistema mais antigo, certifique-se de obter as atualizações mais recentes antes de iniciar o NIS.

Funcionamento do NIS

Os arquivos de dados do NIS são armazenados no diretório **/var/yp**. Cada mapa NIS é armazenado em um formato hashed num subdiretório do diretório NIS nomeado conforme o domínio NIS. Há um arquivo para cada chave pelo qual um arquivo pode ser pesquisado. Por exemplo, no domínio cssuns, os arquivos **gdbm** para os mapas **/etc/passwd** poderiam ser

```
/var/yp/cssuns/passwd.byname
/var/yp/cssuns/passwd.byuid
```

www.snow666.blogspot.com
Snow666

O arquivo **passwd** é pesquisável pelo nome e UID, de modo que dois mapas são derivados dele.

O comando **makedbm** gera mapas NIS de arquivos planos. Entretanto, você não precisa chamar esse comando diretamente. Há um **Makefile** em **/var/yp** que gera todos os mapas NIS comuns. Após modificar um arquivo de sistema, mude (**cd**) para o diretório **/var/yp** e execute **make**. O comando **make** verifica o horário da modificação de cada arquivo comparado com os horários de modificação dos mapas derivados dele e executa **makedbm** para cada mapa que precisa ser reconstruído.

Os mapas são copiados do servidor mestre para os servidores escravos por meio do comando **ypxfr**. **ypxfr** é um comando puxador; ele tem de ser executado em cada servidor escravo para fazer com que aquele servidor importe o mapa. Os escravos normalmente executam **ypxfr** freqüentemente para verificar que realmente estejam de posse dos mapas mais recentes; podemos usar **cron** para controlar com que freqüência isto deve ser feito.

A implementação-padrão da cópia de mapas é um tanto ineficiente. O Linux dispõe de um daemon chamado **ypxfrd** que pode ser executado no servidor mestre para responder de uma forma mais veloz a solicitações feitas por **ypxfr**. **ypxfrd** deixa de lado o protocolo NIS normal e simplesmente envia cópias dos arquivos de mapas.

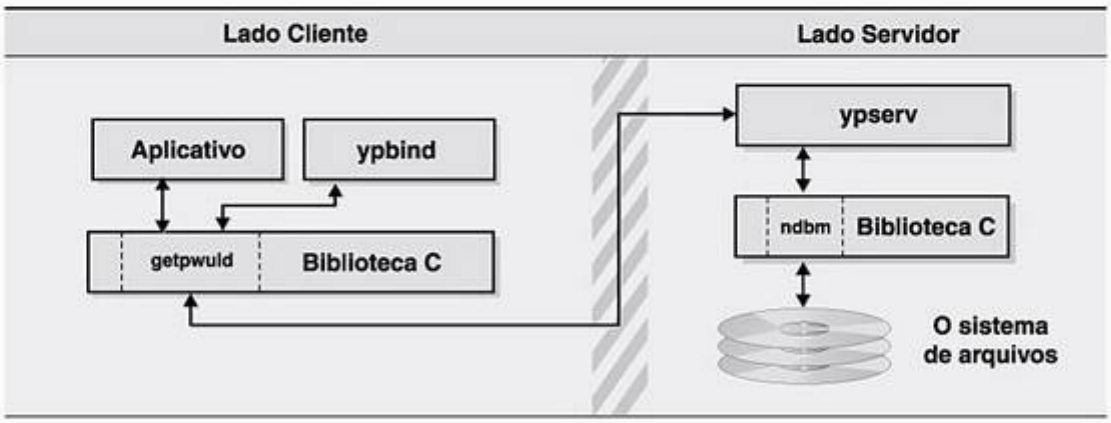
yppush é um comando "empurrão" usado no servidor mestre. Ele, na verdade, não transfere qualquer dado, mas sim instrui cada escravo para executar um comando **ypxfr**. O comando **yppush** é usado por **Makefile** no diretório NIS para garantir que mapas recém-atualizados sejam propagados a escravos.

O mapa especial chamado **ypservers** não corresponde a qualquer arquivo plano. Esse mapa contém uma lista de todos os servidores do domínio. Ele é construído automaticamente quando o domínio é configurado com **ypinit** (veja o tópico *Configurando servidores NIS*, mais à frente). Seu conteúdo é examinado toda vez que o servidor mestre precisar distribuir mapas aos escravos.

Após a configuração inicial, os únicos componentes ativos do sistema NIS são os daemons **ypserv** e **ypbind**. **ypserv** é executado somente nos servidores (tanto mestre quanto escravo); ele aceita consultas de clientes e as responde pesquisando informações nos arquivos de mapas hashed.

ypbind é executado em cada máquina do domínio NIS, inclusive servidores. A biblioteca C contacta o daemon **ypbind** local toda vez que ela precisar responder a uma consulta administrativa (desde que **/etc/nsswitch.conf** solicite para proceder desta maneira). **ypbind** localiza um **ypserv** no domínio apropriado e retorna sua identidade à biblioteca C, que a seguir contacta o servidor diretamente. O mecanismo de consulta é ilustrado no Diagrama A.

Diagrama A Procedimento de consulta no NIS.



Uma vez que **ypbind** localiza um servidor, ele continua a depender deste servidor para todas as consultas até que o servidor saia do ar ou que ocorra algum outro problema de comunicação. Um **ypbind** num servidor não dá a si próprio tratamento preferencial, de modo que os servidores não necessariamente se vinculam a si mesmos.

Em algumas circunstancias (por exemplo, quando todos os servidores, exceto um, são reinicializados simultaneamente), os clientes podem se fixar num servidor e se recusarem a deixá-lo mesmo após outros servidores se tornarem disponíveis. Essa situação pode retardar o tempo de resposta consideravelmente.

O NIS inclui uma série de comandos secundários que examinam mapas, descobrem que versão de um mapa cada servidor está usando e controlam a vinculação entre clientes e servidores. Uma lista completa dos daemons e comandos do NIS é dada na Tabela 18.3. (As linhas foram alternadamente sombreadas para facilitar a leitura.)

Snow666

Tabela 18.3 Comandos e daemons no NIS.

Programa	Descrição
ypserv	O daemon de servidor NIS, iniciado no momento da inicialização
ypbind	O daemon-cliente NIS, iniciado no momento da inicialização
domainname	Configura o domínio NIS no qual uma máquina se encontra (executado no momento da inicialização)
ypxfr	Faz o download da versão atual de um mapa por meio do servidor mestre
ypxfrd	Atende a solicitações de **ypxfr** (executado no servidor mestre)
yppush	Faz com que os servidores escravos atualizem suas versões de um mapa
makedbm	Constrói um mapa hashed por meio de um arquivo plano
ypmake	Reconstrói os mapas hashed por meio de arquivos planos que foram modificados
ypinit	Configura um host como servidor mestre ou escravo
ypset	Faz com que **ypbind** se conecte a um determinado servidor
ypwhich	Descobre que servidor o host atual está usando
yppoll	Descobre que versão de um mapa um servidor está usando
ypcat	Imprime os valores contidos num mapa NIS
ypmatch	Imprime as entradas de mapa para uma chave especificada
yppasswd	Altera uma senha no servidor mestre NIS
ypchfn	Altera as informações GECOS no servidor mestre NIS
ypchsh	Altera um shell de login no servidor mestre NIS
yppasswdd	Servidor para **yppasswd, ypchsh e ypchfn**

Configurando um domínio NIS

O NIS tem de ser inicializado no servidor mestre, nos servidores escravos e em cada cliente. Isso é feito em duas etapas. Em primeiro lugar, execute **ypinit** em cada servidor. A seguir, em cada máquina do domínio, configure o nome de domínio com base em um dos arquivos de inicialização do sistema e configure **/etc/nsswitch.conf** para importar dados NIS.

Configurando servidores NIS

ypinit inicializa tanto o servidor mestre quanto os servidores escravos para um domínio. No mestre, são usados os seguintes comandos.

```
# cd /var/yp          /* O diretório NIS, independentemente de onde ele se encontre */
# domainname foo      /* Atribua um nome ao novo domínio. */
# ypinit -m           /* Inicializa como um servidor mestre. */
# ypserv              /* Inicia o servidor NIS. */
```

O flag **-m** informa **ypinit** que ele está configurando um servidor mestre; ele irá lhe solicitar a introdução de uma lista de servidores escravos. Assim que o mestre estiver ativo e em pleno funcionamento, você deve municiar cada servidor escravo executando **ypinit** com o flag **-s** (slave):

```
# cd /var/yp
# ypinit -s mestre    /* O argumento é o nome de host do mestre. */
# ypserv
```

ypinit -s faz uma cópia local dos dados atuais do mestre; a presença dos arquivos de dados do domínio é suficiente para que **ypserv** saiba que ele deve atender o domínio.

Veja o Capítulo 9 para obter maiores informações sobre ***cron****.*

Em cada escravo, você deve configurar entradas crontab para puxar cópias recentes de todos os mapas do mestre. O comando **ypxfr** *map*, onde *map* é um nome como **passwd.byuid**, transferirá o mapa especificado do servidor mestre. Você tem de executar o comando uma vez para cada mapa. Os mapas tendem a mudar em velocidades diferentes, e se a largura de banda for preciosa, talvez seja interessante transferir alguns mapas com mais freqüência do que outros. Na maior parte das circunstâncias, transferir todos os mapas uma a duas vezes ao dia (quem sabe tarde da noite) já seja suficiente. O script a seguir transfere todos os mapas:

www.snow666.blogspot.com
Snow666


```
#!/bin/sh
meudominio = `/bin/domainname`
cd /var/yp/$mydomain # o diretório NIS
for map in `/bin/ls`; do
      /usr/lib/yp/ypxfr $map
done
```

Também há scripts pré-fabricados em **/usr/lib/yp** que transferem mapas NIS a várias freqüências (**ypxfr_1perday**, **ypxfr_2perday** e **ypxfr_1perhour**).

Se quiser que os usuários sejam capazes de modificar suas senhas com **yppasswd**, você terá de executar o daemon **yppasswdd** no servidor NIS mestre. A versão Linux desse servidor ficou conhecida por travar freqüentemente; portanto, certifique-se de verificar que ela ainda esteja rodando se o comando **yppasswd** parecer não estar funcionando.

Configurando clientes NIS

Após configurar servidores, você tem de informar a cada máquina-cliente que ela é um membro do novo domínio. Os servidores de um domínio geralmente também são clientes.

O comando **domainname** configura o domínio NIS de uma máquina. Normalmente ele é executado no momento da inicialização por meio de um dos scripts de inicialização. Os malabarismos exatos necessários para configurar isto variam conforme a distribuição; detalhes específicos a cada distribuição são apresentados a seguir.

Veja o Capítulo 2 para obter maiores informações sobre os scripts de inicialização.

Cada cliente tem de ter ao menos uma versão privada mínima dos arquivos **passwd**, **group** e **hosts**. **passwd** e **group** são necessários para permitir que root faça o login quando nenhum servidor NIS estiver disponível. Eles devem conter as contas e grupos de sistema padrão: root, bin, daemon etc. O arquivo **hosts** (ou DNS) tem de estar presente para responder a consultas no momento da inicialização que ocorrem antes de o NIS estar ativo e em funcionamento.

No Red Hat, definimos o nome de domínio NIS em **/etc/sysconfig/network** configurando a variável `NISDOMAIN`. **ypbind**, **ypserv** e **yppasswdd** são habilitadas e desabilitadas com **chkconfig**; por exemplo,

```
# chkconfig ypbind on
```

No SuSE, ative **ypbind** editando **rc.config**. Configure `YP_DOMAINNAME` para o nome de seu domínio e altere `START_YPBIND` para `"yes"`. Você também pode (opcionalmente) enumerar uma lista de servidores na variável `YP_SERVER`. Para executar um servidor NIS (**ypserv**), configure `START_YPSERV` em `"yes"`; outros parâmetros de servidor são configurados no arquivo

/etc/rc.config.d/ypserv.rc.config.

O Debian armazena o nome do domínio NIS em **/etc/defaultdomain**. O restante das informações de configuração provém de **/etc/ypbind.config**. O Debian inicia **ypbind** automaticamente caso **/etc/defaultdomain** exista. Você pode transformar o computador num servidor alterando a linha `NISSERVER=false` no arquivo **/etc/init.d/nis** para `true`.

18.4 NIS+: FILHO DE NIS

NIS+, também conhecido como "NIS nos esteróides em cuja face ninguém Jamais Irá Jogar Areia Novamente", foi desenvolvido para corrigir as deficiências do NIS e introduzir deficiências próprias. Ele manipula grandes redes de máquinas. Possui recursos de segurança já incorporados. Permite que vários domínios sejam administrados de qualquer ponto da rede. Transfere atualizações de forma eficiente. É um banco de dados distribuído *e* a cobertura de uma sobremesa; ele canta, dança e ultrapassa a média numa única arrancada.

A Sun lançou o NIS+ no início dos anos 90. Apesar de seus nomes similares, NIS e NIS+ realmente não têm muito a ver um com o outro. O NIS+ é consideravelmente mais complexo do que o NIS e não teve o mesmo grau de apoio popular. A Sun anunciou planos para descontinuar o suporte ao NIS+ e, portanto, nossa expectativa é de que ele vá desaparecer gradualmente ao longo do tempo.

Embora o Linux possa atuar como um cliente NIS+, não há nenhum servidor NIS+ para Linux e ninguém está trabalhando para implementar um. Na qualidade de administrador de sistemas Linux, quase certamente você não terá de lidar com o NIS+ a menos que esteja integrando seus sistemas num ambiente UNIX estabelecido que já o padronizou. Se estiver nesta situação, provavelmente já estará familiarizado com o NIS+.

Snow666

O NIS+ é um bom exemplo do que Frederick P. Brooks, Jr. chama de "o efeito segundo sistema" em seu clássico sobre engenharia de software, *The Mythical Man-Month*. Ele tenta se erguer com o sucesso de seu progenitor e evitando todos os erros e armadilhas do projeto anterior. Ele dedica esforço substancial à arquitetura formal do sistema. Teoricamente isso deveria ser perfeito. Na pratica é um tanto desajeitado, elaborado demais e fora da realidade do dia-a-dia. Fomos informados que até mesmo a Sun não adota o NIS+ internamente.

Sob a perspectiva de um cliente, o NIS+ se parece muito com o NIS. A maior parte dos dados é acessada através das mesmas rotinas de biblioteca como sempre, e o mundo complexo do NIS+ de domínios, tabelas, permissões e caminhos de busca no final é reduzido a um análogo dos arquivos planos originais do UNIX.

Há um cliente NIS+ para Linux, mas ele não é muito bem integrado nas distribuições. Há algumas instruções no *NIS HOWTO* (veja www.linuxdoc.org); elas são um tanto complexas para serem reproduzidas neste livro, porém funcionam moderadamente bem.

O Red Hat inclui o software básico necessário para NIS+, mas nenhuma documentação. Você terá de criar o seu próprio script **rc** para iniciar os daemons necessários. Siga as instruções contidas no *NIS HOWTO*.

Um guia para configuração do NIS+ no ambiente SuSE pode ser encontrado em

http://www.linux-nis.org/nisplus/index.html

Um conjunto de pacotes NIS+ para o Debian se encontra disponível em

http://www.realbodo.de/debian

(Depois disso, siga as instruções contidas no *NIS HOWTO*.)

18.5 LDAP: Lightweight Directory Access Protocol

Os sites UNIX precisam de uma boa maneira para distribuir seus dados de configuração administrativos; porém, o problema é realmente mais abrangente do que isto. Por exemplo, e os dados não administrativos como listas telefônicas e de endereços de e-mail? E as informações que você deseja compartilhar com o mundo externo? O que todo o mundo realmente precisa é de um serviço de diretórios generalizado.

Um serviço de diretórios é apenas um banco de dados, mas um que faz algumas suposições. Qualquer conjunto de dados que possui características que atendam às suposições feitas é um candidato à inclusão. As suposições básicas são as seguintes:

- Objetos de dados são relativamente pequenos.
- O banco de dados será amplamente replicado e colocado em cache.
- As informações se baseiam em atributos.
- Os dados são lidos freqüentemente, mas gravados com pouca freqüência.
- A pesquisa é uma operação comum.

O sistema atual de candidato a padrão do IETF desenvolvido para preencher este papel é o LDAP (Lightweight Directory Access Protocol). As especificações LDAP não falam do banco de dados em si, mas são apenas a maneira pela qual ele é acessado através de uma rede. Mas pelo fato de especificarem como os dados são esquematizados e como as pesquisas são realizadas, eles também implicam um modelo de dados razoavelmente específico.

A história do LDAP é bastante sórdida. Ela começa com o sistema de rede OSI, uma família de protocolos de rede que do início ao fim foi desorientadamente adotada como um padrão internacional em meados dos anos 80. O sistema OSI como um todo demonstrou ser um grande fiasco, mas vários de seus protocolos componentes (o LDAP entre eles) desfrutaram de uma pós-vida macabra em formas mutantes adaptadas à vida do mundo TCP/IP.

O LDAP foi concebido originalmente como um protocolo de gateway simples que permitiria a clientes TCP/IP conversarem com servidores de diretórios X.500 que eram executados em sistemas OSI. Ao longo do tempo, ficou claro que o X.500 estava destinado à morte e também que o UNIX realmente precisava de um diretório-padrão de alguma espécie. Esses fatores levaram o LDAP a ser desenvolvido como um sistema de diretórios completo e independente (e talvez não mais merecendo levar a letra L de lightweight, ou seja, leve).

Até pouco tempo atrás, estávamos em algum ponto no meio do processo de desenvolvimento. A versão mais amplamente implementada do LDAP, a versão 2, é desprovida de vários recursos que serão necessários para que o LDAP atinja o mesmo nível de funcionalidade e de confiabilidade do que, digamos, o DNS. Até mesmo a versão 3 do protocolo, que ainda não está padronizada, parece ter alguns vazios substanciais. Afora alguns poucos domínios específicos (listas telefônicas na Internet, configuração alternativa **sendmail**, alguns aplicativos para agendamento),

Snow666

a experiência prática com o LDAP tem sido limitada. As pessoas estão começando lentamente a usar o LDAP em outros domínios; portanto, com sorte essa situação melhorará.

Infelizmente, o LDAP se tornou uma isca para um frenesi de baixa qualidade que alimenta todo o mercado, com todo o mundo apregoando suporte eterno ao protocolo. Assim como a tecnologia Java em meados dos anos 90, os turbilhões lançaram uma imensidão de press releases, mas relativamente pouco software de verdade. Teremos simplesmente que aguardar e ver se uma bela Vênus eventualmente surge dessas águas.

Nossa percepção sobre o LDAP é que ele pode ou não vir a ser mais desenvolvido no sentido de auxiliar os administradores de sistema. O LDAP realmente ainda não encontrou o seu nicho. Por enquanto recomendamos uma política de "espera vigilante". Isso dito, a maioria das distribuições Linux realmente vêm com suporte para o LDAP como um serviço de diretórios (basta especificá-lo em **nsswitch.conf**), e sites *estão efetivamente* rodando-o no mundo real.

Documentação e especificações do LDAP

Atualmente, a melhor introdução ao LDAP é um livreto chamado *Understanding LDAP* escrito por Heinz Johner et al. para a International Technical Support Organization da IBM. Ele se encontra disponível para download na forma de um arquivo Acrobat em www.redbooks.ibm.com. As partes sobre a API de programação da linguagem C podem ser ignoradas; todo o restante se constitui em informações úteis para os administradores de sistema. E não diga que a IBM jamais fez nada por você.

As RFCs relacionadas com o LDAP são muitas e variadas. Alguns dos pontos-chave são listados na Tabela 18.4. A maioria das RFCs listadas possui equivalentes para a versão 2 que não são mostrados aqui. Como a própria tabela sugere, a maior parte das transações e dos objetos LDAP pode ser representada sob a forma de texto puro, que é um dos melhores recursos do protocolo. É fácil gerar consultas por meio de um script ou configurar gateways para outros protocolos, como o HTTP.

Tabela 18.4 RFCs relacionadas com o LDAP.

RFC	Título
1777	Lightweight Directory Access Protocol (v2)
2251	Lightweight Directory Access Protocol (v3)
2252	LDAPv3: Attribute Syntax Definitions (LDAPv3: Definições da Sintaxe de Atributos)
2253	LDAPv3: UTF-8 String Representation of Distinguished Names (LDAPv3: Representação UTF-8 de Strings de Nomes Distintos)
2254	The String Representation of LDAP Search Filters (A Representação String dos Filtros de Busca LDAP)
2255	The LDAP URL Format (O Formato do URL LDAP)
2256	A Summary of the X.500 User Schema for Use with LDAPv3 (Um Resumo do Esquema de Usuário do X.500 para Uso com LDAPv3)
2307	An Approach for Using LDAP as a Network Information Service (Uma Abordagem para Se Usar LDAP Como um Serviço de Informação de Rede)

A RFC2307 sugere maneiras de mapear conjuntos de dados UNIX tradicionais como os arquivos **passwd** e **group** no espaço de nomes LDAP. Essa RFC ainda está classificada como "experimental", porém é seguida amplamente.

Consulte a seção 19.4 para obter maiores informações sobre como empregar o LDAP com ***sendmail****.*

Mãos na massa com o LDAP

O LDAP foi implementado pela Universidade de Michigan, pela Netscape e por outros. A melhor fonte hoje é o grupo OpenLDAP em www.openldap.org, que assumiu e melhorou o código-base da Universidade de Michigan. O OpenLDAP acompanha o Red Hat, SuSE, Debian e muitas outras distribuições do Linux. São fornecidos alguns documentos, inclusive um guia do administrador (em openldap.org) e um *LDAP HOWTO* (www.linuxdoc.org).

Na distribuição do OpenLDAP, **slapd** é o daemon servidor padrão e **slurpd** manipula a replicação, uma espécie de servidor escravo NIS. Esse esquema acrescenta hierarquia à rede de servidores independentemente de os dados serem verdadeiramente hierárquicos. Quando a versão 3 do protocolo LDAP estiver totalmente desenvolvida, teremos hierarquia verdadeira tanto em termos de dados quanto de infra-estrutura.

www.snow666.blogspot.com
Snow666

Configurando um servidor OpenLDAP

Supondo-se que você tenha instalado o servidor OpenLDAP como parte de sua instalação genérica do Linux, configurá-lo é uma questão de ajuste de alguns arquivos de configuração.

Em primeiro lugar, crie um arquivo **/etc/openldap/slapd.conf**. Um modelo é instalado com o servidor Open-LDAP. Você tem de alterar as seguintes linhas:

```
suffix "dc=meudominio, dc=com"
rootdn "cn=admin, dc=meudominio, dc=com"
rootpw {crypt}abJnggxhB/yWI
```

O `sufixo` é o seu "nome-base LDAP". Ele é a raiz de sua porção do espaço de nomes LDAP, similar conceitualmente a seu nome de domínio DNS. De fato, a prática comum é usar o seu nome de domínio DNS como seu nome-base LDAP (conforme indicado acima). Diferentemente do DNS, o LDAP requer que nomes complexos como meudominio.com sejam subdivididos em componentes de domínio (as entradas `dc=` do exemplo).

`rootdn` é o seu nome de administrador e `rootpw` é a senha formato UNIX (DES) do administrador. Note que os componentes de domínio que levam ao nome do administrador também devem ser especificados. Você pode copiar e colar a senha de **/etc/shadow** (caso não use senhas MD5) ou então gerá-la com um simples script Perl de uma linha

```
perl -e "print crypt('senha','salt');"
```

onde *senha* é a senha desejada e *salt* é uma string arbitrária de dois caracteres. Devido à presença dessa senha, certifique-se de que as permissões em seu arquivo **slapd.conf** sejam 600 e que pertençam ao root.

Também é necessário editar **/etc/ldap.conf** e **/etc/openldap/ldap.conf** para modificar o nome de seu servidor LDAP e de seu nome-base. O processo é bem objetivo — basta configurar o argumento da entrada `HOST para o seu servidor` e configurar `BASE` para o mesmo valor do `sufixo` contido em **slapd.conf**. (Certifique-se de que ambas as linhas não estejam comentadas.) Nesta altura, você deve estar apto a iniciar **slapd** simplesmente executando-o sem nenhum argumento.

No Red Hat, faça com que LDAP seja executado na inicialização habilitando-o com **chkconfig**:

```
# chkconfig ldap on
```

No SuSE, configure LDAP para ser executado no momento da inicialização localizando a entrada `START_LDAP` em **/etc/rc.config** e configurando-a em `"yes"`.

No Debian, a simples instalação do pacote OpenLDAP automaticamente deve fazer com que o servidor inicie no momento da inicialização.

Preenchendo o seu servidor

Você pode preencher o seu banco de dados LDAP pela maneira fácil ou pela maneira difícil. A maneira difícil é fazê-lo manualmente. Entretanto, assim que tiver conhecido um esquema LDAP, você jamais irá querer fazer isto — o LDAP traz o conceito de prolixidade a um novo nível. Por exemplo, uma única entrada de grupo ficaria parecida com o seguinte:

```
dn: cn=staff,ou=groups,dc=nomededominio,dc=com
cn: staff
objectclass: posixGroup
userPassword: {crypt}*
gidNumber: 10
memberuid: root
memberuid: matthew
memberuid: evi
```

Esse é o equivalente LDAP da seguinte linha **/etc/group**:

```
staff:*:10:root,matthew,evi
```

Digite algumas entradas desse tipo, e em breve você estará implorando pelo NIS+. Felizmente, há uma série de ferramentas para gerenciamento de dados LDAP. Particularmente, a Padl Software fornece um conjunto gratuito de scripts em Perl que migram arquivos planos existentes ou mapas NIS para o LDAP. Eles se encontram em www.padl.com/tools.html e estão prontos para execução.

Talvez você encontre alguns problemas; por exemplo, o arquivo **services** padrão não pode ser migrado, pois há sinais + nos nomes de alguns serviços. Isso pode ser corrigido com um pouco de edição, mas mostra mais uma vez

Snow666

quão desafiador é esta coisa de LDAP. Assim que um banco de dados tiver sido importado, podemos verificar se a transferência funcionou corretamente executando o utilitário **slapcat**, que exibe todo o banco de dados.

O LDAP pode ser um sistema extraordinariamente complicado e há muito mais a fazer para embelezar o sistema em termos de segurança, replicação e de outros recursos adicionais. Se você deseja segurança no trabalho, poderá implementar uma configuração LDAP com uma complexidade que desafia a fé de qualquer um. Devido tanto às questões óbvias relativas à manutenção quanto à natureza ainda imatura do LDAP como serviço de diretórios, recomendamos que você mantenha as coisas relativamente simples.

Configurando um cliente LDAP

Assim que tiver um servidor ativo, fica fácil configurar clientes. Edite os arquivos de configuração LDAP (**/etc/ldap.conf** e **/etc/openldap/ldap.conf**) para especificar a identidade do servidor e o nome-base de seu site.

Você poderá verificar se está se conectando corretamente ao servidor LDAP executando **ldapsearch -x**, que faz um dump de todo o banco de dados. O **-x** é importante, pois ele desativa a autenticação.

Finalmente, mude as linhas apropriadas em **/etc/nsswitch.conf** para que use o servidor LDAP como fonte de dados.

LDAP e segurança

Na pior situação, o LDAP contém problemas de segurança suficientes para fazer com que qualquer administrador de segurança em sistemas vagamente consciente estremeça. Em particular (ignorando os problemas de autenticação cliente-servidor), os dados são enviados pela rede em formato de texto puro, de modo que senhas-sombra e outras coisas do tipo não estão protegidas. Recomendamos enfaticamente *não* usar o LDAP para distribuição de senhas-sombra, pelas mesmas razões que não recomendamos distribuí-las usando o NIS. Use **rsync** ou algum outro método.

Por outro lado, há muito mais confiança em relação ao LDAP do que com o NIS. Por exemplo, já que o LDAP é executado numa porta conhecida, você poderia executá-lo via SSL ou até mesmo SSH para fornecer proteção contra bisbilhoteiros. (O suporte SSL para o LDAP está sendo desenvolvido.) Além disso, servidores LDAP (inclusive OpenLDAP) contêm um suporte relativamente granular para ACLs (Access Control Lists, listas de controle de acesso).

Um sistema tão complexo e com tantas partes em movimento como o LDAP inevitavelmente conterá um grande potencial para configuração incorreta acidental que enfraquece a segurança, isto sem mencionar as velhas e conhecidas brechas de segurança. Previna-se, senhor administrador.

18.6 EXERCÍCIOS

E18.1 Por que o **rsync** é considerado mais seguro do que **rdist**?

★ **E18.2** Por que um método "puxão" de atualização dos arquivos de uma máquina local é mais seguro do que um método "empurrão"?

★ **E18.3** Explique o seguinte trecho de um arquivo de controle **rdist**:

```
LINUX_PASSWD = ( redhatbox debianbox susebox )

passwd:
   ( /etc/passwd ) -> ( ${LINUX_PASSWD} )
   install /etc/passwd.rdist;
   cmdspecial /etc/passwd.rdist "/usr/local/sbin/mkpasswd";
```

★ **E18.4** Explique as diferenças entre **rdist** e **rsync**. Em que situações seria melhor usar um em relação ao outro?

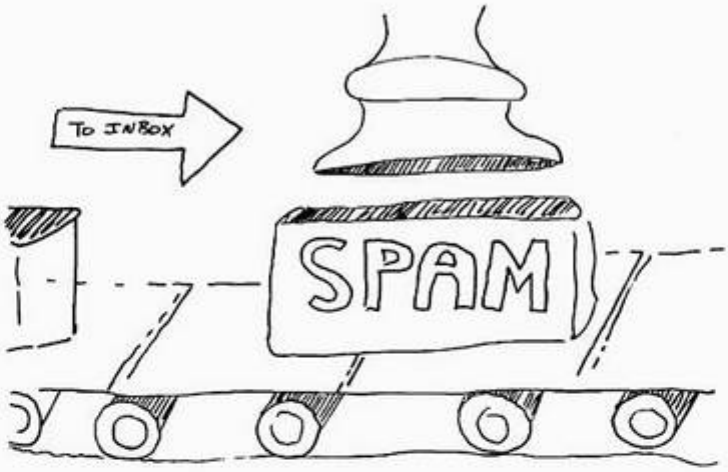
★ **E18.5** Compare e contraste NIS e LDAP. Quando você deveria usar um e não o outro? Alguma vez seria interessante usar ambos?

★ **E18.6** Que método o seu site emprega para compartilhar arquivos de sistema? Quais são os problemas de segurança relacionados com tal método? Sugira uma maneira alternativa para compartilhar arquivos de sistema em seu site e detalhe quais preocupações ele se destina a resolver. Há algum inconveniente?

★★ **E18.7** Descreva as etapas necessárias para assegurar o LDAP usando SSL, o Secure Socket Layer (ou TLS, como é chamado agora; Transport Layer Security).

Snow666

19 E-Mail



Cada vez que tentamos realizar a tarefa de atualizar este capítulo, subestimamos tanto a complexidade de um sistema de e-mail completo em termos de serviços quanto a velocidade com que o software está evoluindo. As principais alterações desde a última vez que reescrevemos este assunto envolviam importantes melhorias no desempenho de centralizadores de e-mail (mail hubs) extremamente congestionados, novos recursos de segurança e ferramentas contra spam.

Hoje, o e-mail é absolutamente essencial na comunicação pessoal e comercial. Muitas das alterações recentes no **sendmail** respondem aos problemas de expansão e de flexibilidade enfrentados pelos ISPs com milhões de clientes famintos por e-mail. O spam inspirou alterações adicionais e reforço cada vez mais rígido das regras existentes. O IETF tem estado atarefado emitindo novas RFCs relacionadas com e-mail. E os hackers têm estado atarefados em invadir qualquer sistema que confie que conteúdo fornecido pelo usuário seja razoável e siga as regras.

Um aspecto social do e-mail que se tornou predominante nos Estados Unidos ao longo dos últimos anos é e-mail comercial não solicitado, coloquialmente chamado de spam, a mala-direta da Internet. Enviar e-mail pela Internet é barato — muito mais barato do que enviar correspondências pelo correio tradicional. Normalmente é cobrado do remetente uma taxa única, de modo que o custo é o mesmo para enviar uma mensagem ou enviar 25 milhões de mensagens. Mas para o provedor Internet (ISP) que deve fornecer largura de banda suficiente para atender ao afluxo de spam — estimado em 30% dos e-mails que chegam na America Online — o spam certamente não é gratuito. Tentativas de controlar spam através de legislações e tribunais tiveram pouco sucesso até então. Soluções técnicas na forma de filtros de e-mails e listas de buraco negro têm sido mais eficientes. Analisamos o estado de arte atual na luta contra o spam na seção 19.10.

O grande volume deste capítulo — quase 90 páginas — atesta a complexidade dos sistemas de e-mail. O capítulo contém tanto informações de formação quanto detalhes de configuração de software, grosseiramente nesta ordem.

Tentamos dividir este capítulo em quatro seções menores (sistemas de e-mail, configuração de **sendmail**, spam e Exim), mas isto o deixaria confuso, cheio de problemas do tipo "quem nasceu primeiro: o ovo ou a galinha?" e, acreditamos, menos útil. Em vez disso, oferecemos a tabela comentada exibida na Tabela 19.1.

Essa organização torna o fluxo um pouco mais tranqüilo quando o capítulo é lido de uma só vez, mas ela algumas vezes separa os itens relevantes a uma determinada tarefa relacionada a e-mail em particular. O postmaster de uma organização mediana precisaria ler o capítulo inteiro, porém um administrador de sistemas configurando o suporte para e-mail em PCs para um cliente típico certamente não teria de fazê-lo.

Snow666

Tabela 19.1 Um guia para este capítulo.

	Seção	Página	Conteúdo
FORMAÇÃO	1	513	Sistemas de e-mail e seus diversos componentes
	2	517	Endereçamento, sintaxe de endereçamento, cabeçalhos de e-mail
	3	523	Filosofia, projeto cliente/servidor, diretórios de armazenamento de e-mail
	4	527	Nomes alternativos, direcionamento de e-mails, software para listas de discussão, LDAP
CONFIGURAÇÃO de **sendmail**	5	541	**sendmail**: instalação, inicialização, fila de e-mail
	6	550	Introdução à configuração de **sendmail**, macros **m4**
	7	554	Primitivas básicas para configuração de **sendmail**
	8	558	Primitivas mais sofisticadas para configuração de **sendmail**
	9	573	Exemplos: máquina doméstica, empresarial, padrões Linux etc.
	10	588	Spam, banco de dados de acesso **sendmail**
	11	603	Segurança
	12	611	Desempenho
	13	615	Coletando estatísticas, testando e depurando
OUTROS	14	620	Exim, uma alternativa para **sendmail** (mais simples, porém não tão flexível)
	15	623	Fontes de informação adicionais

A Tabela 19.2 fornece um guia de navegação para várias das tarefas comuns de administrador de sistemas. A maior parte do capítulo trata da configuração de **sendmail**, o programa-padrão que analisa e direciona e-mails. O **sendmail** foi originalmente escrito por Eric Allman da Universidade da Califórnia, Berkeley. Houve três versões principais: versão 5, IDA e versão 8. A versão 9 está surgindo no horizonte. A versão 5 e o IDA não são mais comumente usadas; a versão 8 as substituiu. Neste capítulo cobrimos a versão 8 (8.12, para ser mais preciso) e antecipamos os recursos esperados na versão 9.

sendmail está sendo desenvolvido comercialmente pela Sendmail, Inc., que também mantém uma versão de código-fonte aberto e gratuita. As versões comerciais dispõem de uma ferramenta gráfica para configuração e geração de relatórios e monitoramento centralizados (recursos particularmente úteis em sites de e-mail de grande volume).

Tabela 19.2 Seções deste capítulo relevantes a várias tarefas.

Atividade	Seções
Atualizando **sendmail**	5, 6
Configurando **sendmail** pela primeira vez	3, 6, 7, 8, 9, 11, 13
Modificando o arquivo de configuração	6
Projetando um sistema de e-mail para um site	3, 4, 6, 7, 8, 9, 11
Combatendo o spam	10
Auditoria na segurança	11
Configurando um PC para receber e-mail	1, 3
Configurando uma lista de discussão	4
Ajustando o desempenho	3, 8, 12
Hospedagem virtual	8, 9
Usando o Exim em vez de **sendmail**	14

19.1 Sistemas de E-Mail

Teoricamente, um sistema de e-mail é formado por quatro componentes distintos:

- Um MUA ("Mail User Agent", "agente de usuário de e-mail") que permite aos usuários lerem e comporem mensagens de e-mail

Snow666

- Um MTA ("Mail Transport Agent", "agente de transporte de e-mails") que direciona mensagens entre as máquinas
- Um "agente para entrega" que coloca as mensagens num armazenador de mensagens local[1]; algumas vezes chamado de LDA (Local Delivery Agent, agente de entrega local)
- Um "agente de acesso" opcional que conecta o agente de usuário ao armazenador de mensagens (por exemplo, por meio de IMAP ou POP)

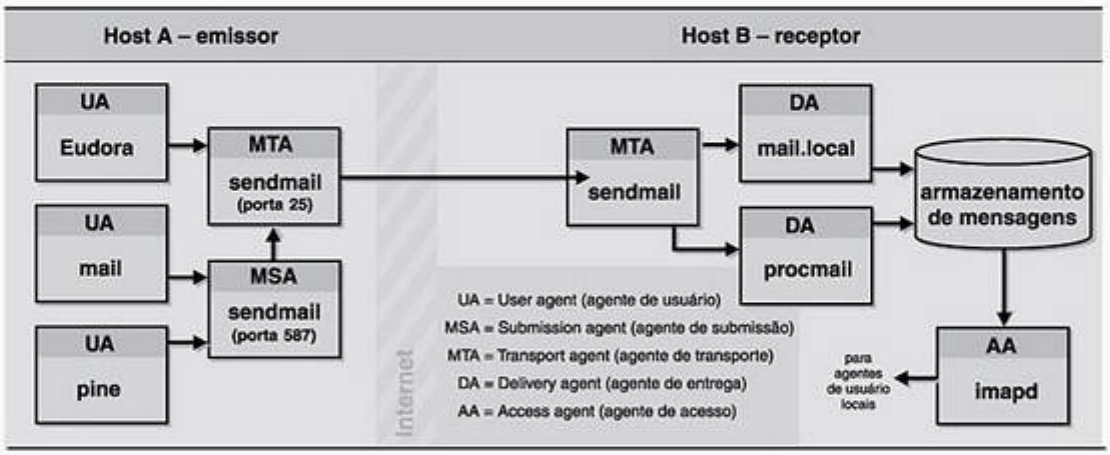
Alguns sites também usam um agente de submissão de e-mails que fala SMTP (o protocolo de transporte de e-mails) e que desempenha a mesma função do agente de transporte. O Diagrama A da página seguinte ilustra o relacionamento entre estes componentes.

Agentes de usuário

Os usuários de e-mail empregam um agente de usuário para ler e compor mensagens. As mensagens de e-mail inicialmente eram formadas somente por texto, porém um padrão conhecido como MIME (Multipurpose Internet Mail Extensions) é agora usado para codificar formatos de texto e anexos (inclusive muitos vírus) no e-mail. Ele é suportado pela maioria dos agentes de usuário. Já que não afeta o endereçamento ou o transporte de e-mail, não trataremos dele neste capítulo.[2]

Uma tarefa dos agentes de usuário é certificar-se de que qualquer texto embutido no conteúdo de uma mensagem de e-mail que possa vir a ser mal interpretada pelo sistema de e-mail se torne protegida. Um exemplo é a string "From " que serve como um separador de registros entre mensagens.

Diagrama A Componentes de um sistema de e-mail



/bin/mail era o agente de usuário original. Hoje existem várias alternativas. Os agentes de usuário populares são listados a seguir, juntamente com suas fontes originais. Distribuições Linux particulares poderão incluir vários desses programas (entre as instalações-padrão dos sistemas usados como exemplo, o SuSE é a mais rica em termos de agentes de usuário e o Debian a mais fraca). Não se aflija se um agente necessário não se encontrar pré-instalado — versões de todos estes pacotes são prontamente disponíveis para o Linux, de modo que você pode instalar facilmente qualquer agente de usuário que seus usuários solicitem.

- **/bin/mail** no Red Hat é a versão BSD do comando **mail** original do UNIX; no SuSE e no Debian, ele se encontra em **/usr/bin/mail**[3]
- **mh** e **nmh** ("novo **mh**") baseado no software da Rand Corporation, mais o front-end **exmh** da Brent Welch of Scriptics
- **pine** da Universidade de Washington, www.washington.edu/pine

1. As caixas postais dos usuários receptores ou, algumas vezes, um banco de dados.
2. Alguns acreditam que o principal efeito do padrão MIME tenha sido o de converter o formato aberto de e-mail da Internet num emaranhado de subformatos proprietários. A poeira ainda está abaixando.
3. Esse agente de usuário algumas vezes é chamado de **Mail** ou **mailx** em outros sistemas. Muito apropriadamente, o Red Hat fornece um link chamado **Mail**, e o SuSE e o Debian fornecem links chamados **Mail** e **mailx**.

Snow666

- **elm** de David Taylor, agora mantido pelo Elm Development Group e por Kari Hurrta, disponível em ftp.virginia.edu
- **mutt** de Michael Elkins, disponível em ftp.mutt.org
- **rmail** e **VM**, agentes de usuário de e-mail que fazem parte do **emacs** e do **XEmacs**; consulte www.wonderworks.com/vm e www.xemacs.org
- Mulberry da Cyrusoft International, Inc., (www.cyrusoft.com/mulberry) para uma série de plataformas
- Messenger da Netscape Communications, para uma série de plataformas
- Eudora da Qualcomm para Macs ou PCs rodando Windows
- Outlook da Microsoft para Windows

Alguns agentes de usuário suportam um arquivo de configuração abrangente que define padrões para todos os usuários. Usuários individuais podem sobrepor esses padrões definindo um arquivo de configuração pessoal em seus diretórios de origem. Os agentes de usuários dos ambientes Windows e Mac são configurados por meio da IU (Interface como Usuário) do aplicativo, embora a maioria também suporte alguma forma de configuração global ou automática para emprego em sites grandes.

A Tabela 19.3 mostra os recursos dos diversos agentes de usuário e as localizações de seus arquivos de inicialização. As localizações e até mesmo os nomes dos arquivos de configuração global são algumas vezes específicos ao fornecedor; muitos se encontram em **/etc**.

Tabela 19.3 Recursos de agentes de usuários e seus arquivos de configuração.

Agente de usuário	Prefs. Sis.	Prefs. Usuário	MIME	POP	IMAP	SMTP
[usr]/bin/mail	**mail.rc**	**.mailrc**				
***mh**	–	**.mh_profile** **.maildelivery**	✓	✓		✓
pine	**pine.conf**	**.pinerc**	✓	✓	✓	✓[a]
elm	**lib/elm.rc**	**.elm/elmrc**	✓	✓[b]	✓[b]	
mutt	**Muttrc**	**.muttrc**	✓	✓	✓	
Mulberry	–	–	✓	✓	✓	✓
Netscape	–	–	✓	✓	✓	✓
Eudora	–	–	✓	✓	✓[c]	✓
Outlook	–	–	3	3	✓	✓

a. **pine** chama **sendmail** na situação-padrão, porém ele também é capaz de falar SMTP.
b. Não suportado na versão-padrão, porém uma correção se encontra disponível.
c. Incluso na versão comercial, mas não na versão gratuita.

A coluna SMTP se refere à maneira pela qual o agente de usuário transmite e-mail para o agente de transporte ou para o agente de submissão. Uma marca de verificação significa que o agente de usuário abre uma conexão de rede diretamente com o agente de transporte ou de submissão. Nenhuma marca significa que o agente de usuário executa o agente de transporte ou de submissão na forma de um subcomando.

Agentes de transporte

Um agente de transporte deve aceitar e-mail de um agente de usuário, compreender os endereços dos receptores e de alguma maneira mandar o e-mail para os hosts corretos para entrega. A maioria dos agentes de transporte também atua como agente de submissão de mensagens para a introdução inicial de novas mensagens no sistema de e-mail. Os agentes de transporte falam SMTP (Simple Mail Transport Protocol, protocolo de transporte de e-mail simples) definido na RFC2821 (originalmente definido na RFC821) ou o protocolo ESMTP (Extended SMTP, SMTP Estendido) definido nas RFCs 1869, 1870, 1891 e 1985.

Vários agentes de transporte estão disponíveis para os sistemas UNIX e Linux (PMDF, Postfix, **smail**, **qmail**, Exim e **zmailer**, entre outros), porém **sendmail** é o mais abrangente, o mais flexível e o mais amplamente usado. Uma pesquisa recente[4] de sistemas de e-mail relatou que o **sendmail** era usado por cerca de 60% dos domínios, Exim por 8%, Microsoft Exchange Server por 4% e o Postfix por 2%. Os demais (cerca de 50 deles) eram irrelevantes.

4. Estudo privado realizado pela Matrix.net para a Sendmail, Inc.

Snow666

O Red Hat e o SuSE já vêm com **sendmail** instalado. O Debian diz incluir **sendmail**, porém, se você observar com cuidado descobrirá que **sendmail** é, na realidade, um link para o agente de transporte Exim. O Exim foi cuidadosamente trabalhado para compreender os flags de linha de comando de **sendmail**. Os agentes de usuário que chamam "**sendmail**" explicitamente para submeterem e-mail não perceberiam a diferença.

Agentes de entrega

Um agente de entrega aceita e-mail de um agente de transporte e, na verdade, o remete aos receptores locais apropriados. O e-mail pode ser entregue para uma pessoa, para uma lista de discussão, para um arquivo ou até mesmo para um programa.

Cada tipo de receptor pode requerer um agente diferente. **/bin/mail** é o agente de entrega para usuários locais. **/bin/sh** é o agente de entrega original para e-mail direcionados para um programa; entrega para um arquivo é manipulada internamente. Versões recentes de **sendmail** vêm com agentes de entrega locais mais seguros chamados **mail.local** e **smrsh** (cuja pronúncia é "smursh"). O **procmail** da www.procmail.org também pode ser usado como um agente de entrega local; veja a seção 19.8.

Armazenamento de mensagens

À medida que o e-mail cresceu saindo do atendimento a uma comunidade universitária dos departamentos de computação para passar a atender sites como o America Online com milhões de assinantes, o sistema de arquivos se tornou inadequado como armazenador de mensagens. Pesquisar um diretório contendo milhões de caixas postais é proibitivo.

O armazenador de mensagens é o ponto na máquina local em que as mensagens de e-mail são armazenadas. Ele costumava ser o diretório **/var/spool/mail** ou **/var/mail**, com os e-mails sendo armazenados em arquivos cujos nomes se baseavam nos nomes de login dos usuários. Este ainda é o tipo de armazenador de mensagens mais comum, porém os ISPs com milhares ou milhões de clientes de e-mail estão procurando outras tecnologias para o armazenamento de mensagens (bancos de dados, normalmente).

Em sistemas que usam o armazenador **/var/spool/mail** ou **/var/mail**, o diretório para e-mail é criado durante a instalação do sistema operacional. Ele deve ter permissões configuradas para o modo 775, com o dono do grupo mail[5], a menos que você use **mail.local** como seu programa de entrega local, em cujo caso o modo pode ser 755. Em plataformas Linux isso varia um pouco:

```
Red Hat:  drwxrwxr-x 2 root mail 1024 Dec 5 2000   /var/spool/mail
SuSE:     drwxrwxrwt 2 root root 4096 Aug 2 23:25  /var/spool/mail
Debian:   drwxrwsr-x 2 root mail 4096 Aug 3 16:17  /var/mail
```

Consulte a página 45 para obter maiores informações sobre o bit sticky.

As permissões do SuSE são um tanto generosas, contudo os arquivos dentro do diretório de spool[6] de e-mails são modo 660 com grupo root. Diretórios com o bit sticky ativo (o `t` nas permissões) não permitem que usuários eliminem arquivos de outros usuários muito embora eles tenham permissão de gravação no diretório. Porém, um usuário mal-intencionado poderia preencher o spool de e-mail, e usá-lo como uma partição de rascunho ou criar a caixa postal de um outro usuário.

Agentes de acesso

Programas como **imapd** e **spop** são agentes de acesso para usuários PC, Mac ou Linux cujo e-mail é entregue a um servidor Linux e depois descarregado, respectivamente, com o IMAP (Internet Message Access Protocol) ou com o POP (Post Office Protocol). O IMAP e o POP são vistos inicialmente na seção 19.3.

Agentes de submissão de e-mails

Um novo participante da arena de sistemas de e-mail que era requerido em sites com grande volume de e-mails é o agente de submissão. O agente de transporte num centralizador de e-mail congestionado perde muito tempo pré-processando mensagens de e-mail: garantindo que todos os nomes de hosts estejam totalmente qualificados, modificando cabeçalhos herdados de agentes de usuários de e-mail defeituosos, registrando mensagens de erro, reescrevendo cabeçalhos e assim por diante. A RFC2476 introduziu a idéia de separar o MSA (Mail Submission Agent,

5. Sistemas que entregam e-mails criando arquivos com um **chown** diferente de root também têm de ter permissão de gravação de grupo no diretório. Em geral, **chown** difrente de root é uma má idéia.
6. Normalmente para armazenamento temporário. Utilizado também pelos servidores de impressão enquanto enviam os trabalhos submetidos para a(s) impressora(s) (N.R.T.).

Snow666

agente de submissão de e-mail) do MTA (Mail Transport Agent, agente de transporte de e-mail) para distribuir a carga de trabalho e maximizar o desempenho.

A idéia é usar o MSA, que roda numa porta diferente, como uma espécie de "recepcionista" para novas mensagens introduzidas no sistema de e-mail por agentes de usuário local. O MSA realiza todo o trabalho preparatório e de verificação de erros que deve ser feito antes de uma mensagem poder ser enviada pelo agente de transporte. É como se estivéssemos inserindo um verificador de sanidade entre o MUA e o MTA.

Em particular, o MSA garante que todos os nomes de hosts estejam totalmente qualificados; ele verifica se os nomes de hosts são legítimos antes de acrescentar a porção de domínio local. O MSA também corrige cabeçalhos de mensagens caso estejam faltando ou não atendam aos padrões. Normalmente, o MSA adiciona um cabeçalho From ou Date ou ajusta o cabeçalho Message-Id. Uma tarefa final que um MSA pode fazer é reescrever o endereço do remetente com base em um nome de login para uma forma externa preferencial como *Primeiro_Último*.

Para que esse esquema funcione, os agentes de usuário têm de ser configurados para se conectarem ao MSA da porta 587 em vez da porta 25, que é a porta tradicional para e-mail. Se seus agentes de usuário não forem capazes de usar a porta 587, você ainda pode executar um MSA na porta 25, porém num outro sistema que não seja aquele que executa o seu MTA. O agente de transporte também deve ser configurado para que ele não duplique o trabalho feito pelo MSA. Duplicar processamento não afetará a correção da manipulação de e-mails, mas representa efetivamente um trabalho extra inútil.

sendmail pode atuar como um MSA, bem como um MTA. Iniciando com o **sendmail** 8.10, uma única instância do programa escuta tanto na porta 25 quanto na porta 587. Os agentes de usuário normalmente chamam **sendmail** diretamente com flags que solicitam a ele para aceitar uma mensagem de e-mail (**-bs** ou **-bm**) ou sem usar nenhum flag, em cujo caso o comportamento de **sendmail** adotará como padrão **-bm**. O processo de **sendmail** registra a forma como ele foi chamado e se torna um MSA se chamado com flags **-bs** ou **-bm** ou um MTA se chamado com **-bd**.

Os agentes de usuário que abrem diretamente uma conexão SMTP têm de ser modificados para usar a porta 587 para tirar proveito de um MSA.

19.2 A Anatomia de uma Mensagem de E-Mail

Uma mensagem de e-mail apresenta três partes distintas que precisamos entender antes de nos envolvermos com a configuração de sendmail.

- O envelope
- Os cabeçalhos
- O corpo da mensagem

O envelope determina onde a mensagem será entregue ou, se a mensagem não puder ser entregue, para quem ela deveria ser retornada. Os endereços do envelope geralmente concordam com as linhas From e To do cabeçalho para um único receptor, mas não concordam se a mensagem for enviada para uma lista de discussão. Os endereços são fornecidos separadamente para o MSA. O envelope é invisível para os usuários; ele é usado internamente por **sendmail** para descobrir para onde enviar a mensagem.

Os cabeçalhos são um conjunto de pares propriedade/valor formatados de acordo com a RFC2822. Eles registram uma série de informações sobre a mensagem, como a data e o horário em que ela foi enviada e os agentes de transporte através dos quais ela percorreu sua jornada. Os cabeçalhos são uma parte autêntica da mensagem de e-mail, embora agentes de usuário normalmente ocultem alguns dos menos interessantes ao exibir mensagens para o usuário.

O corpo da mensagem é o conteúdo efetivo a ser enviado. Ele deve ser formado por texto puro, embora este texto normalmente represente uma codificação segura para e-mail de vários conteúdos binários.

Quando ingressarmos na seção sobre configuração, algumas vezes falaremos do remetente e dos receptores do envelope e outras vezes do remetente e dos receptores do cabeçalho. Tentamos especificar quais endereços estamos nos referindo caso não fique claro pelo contexto.

Endereços de e-mails

O endereçamento local é simples, pois o nome de login de um usuário é um identificador exclusivo. Um endereço Internet também é simples: *usuário@host.domínio* ou *usuário@domínio*. Nos primórdios do e-mail e da Internet, endereços como os mostrados na Tabela 19.4 eram comuns.

www.snow666.blogspot.com
Snow666

Tabela 19.4 Exemplos de tipos de endereços obsoletos.

Tipo de endereço	Exemplo de endereço	Forma moderna
UUCP	mcvax!uunet!ucbvax!hao!boulder!lair!evi	evi@lair
Baseado em rotas	<@site1,@site2,...,@siteN:user@final-site>	user@final.site
"Notação com o símbolo de Porcentagem"	user%host1%host2@host3	user@host1

Grande parte da complexidade da configuração de **sendmail** provém da exigência antiga de manipular tais endereços. Cada uma dessas formas de endereçamento depende de retransmissão e, graças aos spammers, os sites estão lentamente desativando a retransmissão. A notação com o símbolo de porcentagem (última linha da Tabela 19.4) é uma ferramenta favorita dos spammers que tentam ocultar suas identidades ou para retransmitir e-mail por meio da sua máquinas. Se você precisar lidar com qualquer um dessas formas de endereço, veja a documentação de **sendmail** ou o livro da O'Reilly sobre **sendmail** para uma ajuda.

Lendo os cabeçalhos de mensagens de e-mail

Toda mensagem começa com várias linhas chamadas de cabeçalhos que contêm informações sobre a mensagem. Cada cabeçalho começa com uma palavra-chave como To, From ou Subject, seguida por dois-pontos e o conteúdo do cabeçalho. O formato dos cabeçalhos-padrão é definido na RFC2822; entretanto, também são permitidos cabeçalhos personalizados. Qualquer cabeçalho começando com "X-" é ignorado pelo sistema de e-mail, mas é propagado juntamente com a mensagem. Portanto, você pode adicionar um cabeçalho como X-Pegadinha-do-Dia a suas mensagens de e-mail sem interferir na capacidade do sistema de e-mail de encaminhá-las.[7]

Alguns cabeçalhos são acrescentados pelo agente de usuário e outros pelo agente de transporte. Vários cabeçalhos rastreiam o caminho de uma mensagem através do sistema de e-mail. Muitos dos agentes de usuário ocultam esses cabeçalhos "desinteressantes" de você, mas usualmente há uma opção para fazer com que o agente os revele. Ler cabeçalhos está se tornando uma habilidade importante à medida que somos bombardeados com spam e temos algumas vezes de tentar rastrear uma mensagem em seu caminho inverso até chegarmos à sua origem. Eis o bloco de cabeçalho de uma mensagem simples:

```
From evi Wed Jan 19 19:01:11 2000
Received: (from evi@localhost) by xor.com (8.9.3/8.9.3) id TAA17820; Wed, 19
   Jan 2000 19:01:11 -0700 (MST)
Date: Wed, 19 Jan 2000 19:01:11 -0700 (MST)
From: Evi Nemeth <Evi.Nemeth@xor.com>
Message-Id: <200001200201.TAA17820@xor.com>
To: trent@xor.com
Subject: xor.mc
Cc: evi@xor.com
Status: R

----- aqui entrava o corpo da mensagem ---
```

Esta mensagem permaneceu completamente na máquina local; o remetente era evi e o receptor era trent. A primeira linha From foi acrescentada por **mail.local** que, nesse caso, era o agente de entrega local. As linhas de cabeçalho Subject e Cc foram acrescentadas pelo agente de usuário de e-mail de evi, que provavelmente também adicionou os cabeçalhos To, From e Date. O cabeçalho Status foi adicionado pelo programa para leitura de e-mails de trent. **sendmail**, o agente de transporte de e-mails, acrescentará os cabeçalhos To, From e Date caso eles não sejam fornecidos por MUA. Cada máquina (ou, mais precisamente, cada MTA) que tem contato com uma mensagem adiciona um cabeçalho Received.

Os cabeçalhos numa mensagem de e-mail fornecem uma grande quantidade de informações relativas aos locais por onde a mensagem esteve, por quanto tempo ela permaneceu nesses locais e quando ela foi finalmente entregue ao seu destinatário. A seguir apresentamos uma análise mais completa de uma mensagem de e-mail enviada pela Internet. Ela é entremeada com comentários que descrevem o propósito dos diversos cabeçalhos e identificam os

7. Tecnicamente falando, podemos adicionar qualquer cabeçalho que desejarmos, pois o roteamento de e-mails usa somente o envelope e ignora os cabeçalhos.

www.snow666.blogspot.com
Snow666

programas que os adicionaram. Os números de linhas à esquerda são para referência na discussão que segue e não fazem parte da mensagem. Algumas linhas foram reduzidas para permitir que o exemplo coubesse na página.

```
1: From eric@knecht.sendmail.org
```

A linha 1 foi adicionada por **/bin/mail** ou **mail.local** durante a entrega final para separar essa mensagem das demais na caixa postal do usuário receptor. Alguns programas para leitura de e-mails reconhecem os limites da mensagem procurando uma linha em branco seguida pelos caracteres "From "; observe o espaço no final. Essa linha não existe até que a mensagem seja entregue e ela é distinta da linha de cabeçalho "From:". Muitos programas de leitura de e-mails não exibem essa linha, de modo que você poderá não vê-la.

```
2: Return-Path: eric@knecht.Neophilic.COM
```

A linha 2 especifica um caminho de retorno, que pode ser um endereço diferente daquele exibido na linha From: posteriormente no cabeçalho do e-mail. Mensagens de erro devem ser enviadas à linha de cabeçalho Return-Path; ela contém o endereço do remetente do envelope.

```
3: Delivery-Date: Mon, 06 Aug 2001 14:31:07 -0600
```

A linha 3 mostra a data em que o e-mail foi entregue para a caixa postal local de evi. Ela inclui a diferença de fuso horário entre o horário UTC e o horário local (MDT, mountain daylight time).

```
4: Received: from anchor.cs.colorado.edu (root@anchor.cs.colorado.edu
   [128.138.242.1]) by rupertsberg.cs.colorado.edu (8.10.1/8.10.1) with ESMTP
   id f76KV7J25997 for <evi@rupertsberg.cs.colorado.edu>; Mon, 6 Aug 2001
   14:31:07 -0600 (MDT)
5: Received: from mroe.cs.colorado.edu (IDENT:root@mroe.cs.colorado.edu
   [128.138.243.151]) by anchor.cs.colorado.edu (8.10.1/8.10.1) with ESMTP id
   f76KV6418006 for <evi@anchor.cs.colorado.edu>; Mon, 6 Aug 2001 14:31:06
   -0600 (MDT)
6: Received: from knecht.Neophilic.COM (knecht.sendmail.org [209.31.233.176])
   by mroe.cs.colorado.edu (8.10.1/8.10.1) with ESMTP id f76KV5Q17625 for
   <evi@anchor.cs.colorado.edu>; Mon, 6 Aug 2001 14:31:05 -0600 (MDT)
7: Received: from knecht.Neophilic.COM (localhost.Neophilic.COM [127.0.0.1])
   by knecht.Neophilic.COM (8.12.0.Beta16/8.12.0.Beta17) with ESMTP id
   f76KUufp084340 for <evi@anchor.cs.colorado.edu>; Mon, 6 Aug 2001 13:30:
   56 -0700 (PDT)
```

As linhas 4 a 7 documentam a passagem da mensagem através dos diversos sistemas no percurso para a caixa postal do usuário. Cada máquina que manipula uma mensagem de e-mail acrescenta uma linha Received ao cabeçalho da mensagem. Novas linhas são adicionadas na parte superior, para que ao lê-las você estará rastreando a mensagem do receptor pelo caminho inverso ao remetente. Se a mensagem que você estiver olhando for um spam, a única linha Received em que você realmente poderá acreditar é aquela gerada pela sua máquina local.

Cada linha Received inclui o nome da máquina remetente, o nome da máquina receptora, a versão de **sendmail** (ou seja lá qual for o agente de transporte usado) na máquina receptora, o identificador exclusivo da mensagem enquanto na máquina receptora, o receptor (se houver somente um), a data e o horário e, finalmente, a diferença de fuso horário entre o horário UTC (Universal Coordinated Time, anteriormente chamado de GMT, de Greenwich Mean Time) e a zona de horário local. Estes dados são coletados das variáveis de macro internas de **sendmail**. Nos próximos parágrafos, rastreamos a mensagem do remetente ao receptor (percorrendo o caminho inverso, sob o ponto de vista das linhas de cabeçalho).

Consulte a página 311 para obter maiores informações sobre registros MX.

A linha 7 mostra que a mensagem foi da interface localhost de knecht (cujo agente de usuário de e-mail particular de Eric escolheu para sua conexão inicial) para a interface externa de knecht através do pseudodispositivo de loopback de kernel. A linha 6 documenta o fato de que knecht envia depois a mensagem para mroe.cs.colorado.edu, muito embora a mensagem tivesse sido endereçada a evi@anchor.cs.colorado.edu (veja o cabeçalho da linha 9). Uma rápida verificação com **dig** ou **nslookup** mostra que o host anchor possui um registro MX que aponta para mroe, fazendo com que a entrega seja desviada. A máquina knecht estava usando o **sendmail** versão 8.12.0Beta16.

A máquina mroe estava usando **sendmail** versão 8.10.1 e ela identificou a mensagem com a fila de ID f76KV5Q17625 enquanto se encontrava lá. A seguir mroe encaminhou a mensagem para anchor.cs.colorado.edu conforme endereçado (linha 5), que pode parecer estranho dado que a transmissão original de knecht se desviou de anchor para mroe devido aos registros MX. A razão para essa aparente inconsistência é que o domínio cs.colora-

www.snow666.blogspot.com
Snow666

do.edu usa uma configuração "DNS dividida". O registro MX para anchor que é visível ao mundo externo aponta para a máquina mestra para e-mails que chegam (mroe). Entretanto, um registro diferente é visto dentro do próprio domínio cs.colorado.edu. A versão interna do registro aponta primeiramente para o próprio anchor e depois para mroe como um backup.

Assim que o e-mail chegou em anchor, ele foi imediatamente encaminhado novamente, dessa vez para rupertsberg. A causa desse salto foi o emprego de nomes alternativos, um recurso para manipulação de e-mails que é descrito em detalhes a partir da seção 19.4.

Os nomes alternativos desempenham um papel importante no fluxo de e-mails. Um nome alternativo associa um nome de usuário a uma outra coisa; por exemplo, para o mesmo usuário numa máquina diferente, para um grupo de usuários, ou até mesmo para uma forma de escrita alternativa do nome de usuário. Você não pode determinar qual mensagem foi desviada examinando somente os cabeçalhos de exemplo. Como acontece com os registros MX, temos de procurar fontes de informação externas.

As linhas recebidas 5 e 6 incluem a frase "for <evi@anchor.cs.colorado.edu>", que identifica como o e-mail foi endereçado quando ele chegou ao site local. Essa informação é muito útil quando tentamos retirar a inscrição de uma lista de discussão que requer que enviemos a mensagem de retirada de inscrição ("unsubscribe") a partir do mesmo host que fizemos a assinatura (algumas vezes anos atrás), ou então para conhecer esse endereço e usá-lo como um parâmetro em nossa mensagem de retirada de inscrição.

A linha Received final (linha 4) mostra "for <evi@rupertsberg.cs.colorado.edu>"; o valor da macro de endereço de destino de **sendmail** foi alterado pela pesquisa de nome alternativo na máquina anchor. O agente de entrega de e-mails local em rupertsberg colocou o e-mail na caixa postal de evi.

```
8: Message-Id: <200108062030.f76KUufp084340@knecht.Neophilic.COM>
```

A linha 8 contém o ID da mensagem, que é diferente de um ID de fila e é exclusivo dentro do sistema de e-mail mundial. Ele é adicionado à mensagem quando ela é submetida inicialmente ao sistema de e-mail.

```
9: To: evi@anchor.cs.colorado.edu
10: From: Eric Allman <eric@Sendmail.ORG>
11: X-URL: http://WWW.Sendmail.ORG/~eric
12: Subject: example message for Evi
13: Date: Mon, 06 Aug 2001 13:30:56 -0700
14: Sender: eric@knecht.Neophilic.COM
```

As linhas 9, 10, 12, 13 e 14 são padrão. Embora um cabeçalho Subject (Assunto) não seja exigido, a maioria dos agentes de usuário o inclui. A linha To (Para) contém o endereço do receptor ou receptores primários. A linha From (De) lista o remetente como eric@sendmail.org; entretanto, as linhas Received listam a máquina remetente como estando no domínio neophilic.com — a máquina knecht de Eric possui vários domínios virtuais presos a ele além de sendmail.org.

A linha Date (Data e hora) mostra a data e o horário em que a mensagem foi enviada. Nesse caso o horário de envio quase coincide com as datas nas linhas Received (Recebida em), muito embora cada um deles fosse medido com um relógio diferente.

A linha 11 identifica o URL da homepage de Eric. Note que ela começa com um X, tornando-a um cabeçalho não oficial. Quando o e-mail foi especificado pela primeira vez, não havia coisas como a Web ou os URLs.

As linhas Received normalmente são acrescentadas pelo agente de transporte (a menos que elas sejam forjadas) e os demais cabeçalhos são acrescentados pelo agente de usuário. Alguns agentes de usuário são incorretos e não adicionam os cabeçalhos apropriados; nesse caso, **sendmail** entra em ação para adicionar os cabeçalhos faltantes.

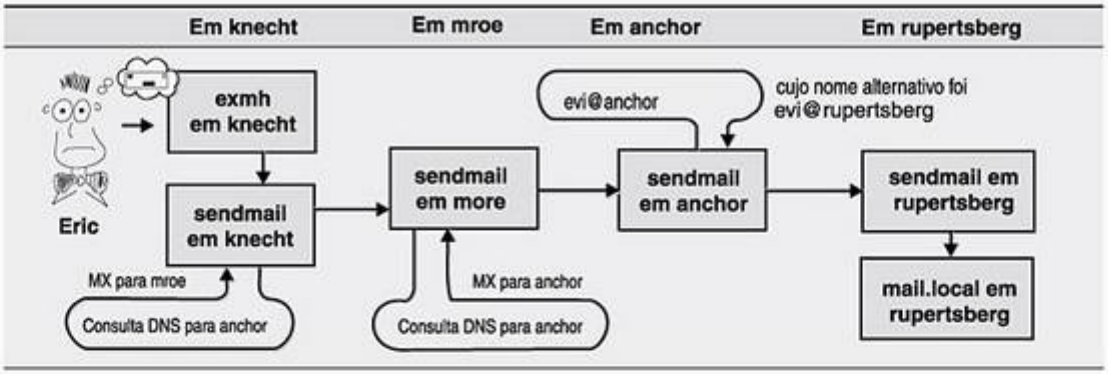
A primeira linha Received, adicionada (normalmente na máquina remetente, quando o e-mail é transferido para a interface de saída), algumas vezes inclui uma estrutura "ident" que dá o nome de login do remetente. Ele deve ser o mesmo que o nome encontrado na linha From, porém não será se a linha From for forjada. Em nosso exemplo, a máquina knecht de Eric não estava executando o daemon que implementa este recurso (**identd**), de modo que não há nenhuma estrutura que liste o nome de login do remetente.

O Diagrama B ilustra a jornada percorrida por essa mensagem através do sistema de e-mail. Ele mostra quais medidas foram tomadas, onde elas aconteceram e quais programas as realizaram. Como pode ser observado, **sendmail** é o burro de carga nesse processo. Ele manipula a mensagem desde o momento em que ela deixa **exmh** em Berkeley até ela chegar a rupertsberg para entrega.

Veja o tópico *Exemplos de spam* na seção 19.10 para ganhar mais prática na leitura de cabeçalhos.

Snow666

Diagrama B Uma mensagem de Eric



19.3 Filosofia do E-Mail

A filosofia de e-mail que descrevemos neste capítulo é quase que obrigatória para manter gerenciável a administração de sites de médio e grande porte. Entretanto, ela também se aplica a sites pequenos. Os principais conceitos que levam a uma administração mais tranqüila são os seguintes:

- Servidores para e-mail que são recebidos e enviados; ou, para sites realmente grandes, uma hierarquia
- Uma central de e-mail ("mail home") para cada usuário num site físico
- IMAP ou POP[8] para integrar clientes remotos, PCs e Macs

Consulte a seção 16.11 para obter maiores informações sobre registros MX.

Discutiremos cada uma dessas questões fundamentais a seguir e depois daremos alguns exemplos. Outros subsistemas também têm de cooperar com o projeto de seu sistema de e-mail: os registros MX do DNS têm de ser configurados corretamente, firewalls de Internet devem permitir a entrada e a saída de e-mails, a(s) máquina(s) para armazenamento de mensagens deve(m) ser identificada(s) e assim por diante.

Os servidores de e-mail têm quatro funções:

- Aceitar e-mails que saem de agentes de usuários e introduzi-los no sistema de e-mail
- Receber e-mails provenientes do mundo externo
- Entregar e-mails nas caixas postais dos usuários finais
- Permitir que os usuários acessem suas caixas postais com IMAP ou POP

Num site pequeno, os servidores que implementam essas funções poderiam todos estar na mesma máquina preenchendo funções diferentes. Em sites maiores, eles devem ser máquinas distintas. É muito mais fácil configurar as regras de firewall de uma rede se os e-mails recebidos chegarem somente numa máquina e os que são enviados parecer se originar de somente uma máquina.

Alguns sites usam um proxy para receber e-mails do mundo externo. O proxy realmente não processa e-mails; ele simplesmente os aceita e os joga em spools. Um processo separado encaminha então o e-mail em spool para **sendmail** para transporte e processamento. **smtpd** e **smtpfwdd** de www.obtuse.com são exemplos de tais proxies para **sendmail**; **smtpd** também pode filtrar e-mail recebido com listas de acesso. Ambos são produtos de fonte aberto. Nenhuma das distribuições Linux usadas como exemplo neste livro os inclui no pacote de instalação padrão.

Usando servidores de e-mail

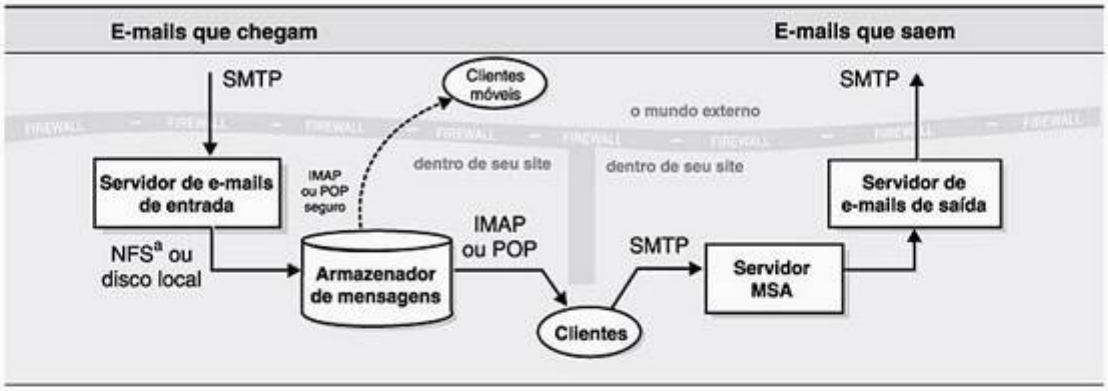
Escolha máquinas estáveis e confiáveis para seus servidores de e-mail. Aqui, descrevemos o projeto de um sistema de e-mail que parece ser expansível, seguro e relativamente fácil de ser gerenciado. Ele centraliza a manipulação tanto de e-mails recebidos como enviados em servidores dedicados para estes propósitos. O Diagrama C ilustra uma forma desse sistema.

O sistema de e-mail representado no Diagrama C possui um único ponto de exposição para o mundo externo: o servidor de e-mail que recebe mensagens da Internet. O servidor de e-mails que saem também está conectado diretamente com a Internet, porém ele é menos exposto, pois inicia conexões em vez de aceitar conexões de sites externos. O servidor de e-mails que chegam deve ser monitorado cuidadosamente, deve ser atualizado com correções para problemas de segurança e deve rodar a última versão de **sendmail** com filtros contra spam para e-mail que chega.

8. Hoje o IMAP é preferido em relação ao POP.

Snow666

Diagrama C Arquitetura de sistema de e-mail



a. Não recomendamos o emprego de NFS para o armazenamento de mensagens devido à probabilidade de geração de problemas de bloqueio de arquivos.

O servidor que manipula e-mails enviados também tem de ter uma boa manutenção. Ele pode incluir filtros de spam próprios para verificar se nenhum usuário local está contribuindo para o problema de spam. Se seu site tiver preocupações quanto a vazamento de informações proprietárias, estabelecer um único servidor através do qual todos os e-mails enviados devem passar facilita a implementação ou reforça as políticas de conteúdo. Se seu site controla grandes listas de discussão, o servidor para e-mails enviados pode ser configurado para tirar proveito de alguns recursos orientados ao desempenho de **sendmail**; veja a seção 19.12 para maiores detalhes.

Tanto os servidores de e-mails recebidos quanto enviados podem ser replicados caso sua carga de e-mails exija isto. Por exemplo, vários servidores para e-mails recebidos podem ocultar na retaguarda um computador para balanceamento de carga ou usar registros MX do DNS para balancear grosseiramente a carga. Entretanto, máquinas-cliente diferentes são capazes de direcionarem e-mail através de servidores para envio. Porém, não passe qualquer e-mail diretamente entre os servidores de entrada e de saída; eles devem ser separados uns dos outros por um firewall interno.

Em sites realmente grandes, os servidores de entrada e de saída devem ser replicados. Uma camada adicional de roteamento poderia ser adicionada para pesquisar as caixas postais dos usuários (talvez via LDAP) e direcionar os e-mails para o armazenador de mensagens apropriado. A camada de roteamento também poderia executar filtragem de spam e de vírus antes de entregar mensagens às caixas postais dos usuários.

Os ISPs (provesores de acesso e outros serviços) que estão projetando um sistema de e-mails para clientes deveriam adicionar um outro servidor que atuaria como o destino dos registros MX de backup dos clientes e manipularia as listas de discussão. Essa máquina tem de aceitar e-mails e retransmiti-los, mas tem de ser altamente filtrada para garantir que retransmita somente os e-mails de clientes verdadeiros. Ela também deveria ser separada dos servidores de e-mail de entrada e de saída por um firewall.

Consulte a seção 18.1 para uma discussão sobre questões referentes à distribuição de arquivos.

Os hosts Linux comuns podem receber uma configuração mínima para **sendmail** que encaminhe, para processamento, e-mails de saída para o servidor. Eles não precisam aceitar e-mails da Internet. Alguns sites poderão querer relaxar um pouco esse modelo de afunilamento e permitir que hosts arbitrários enviem e-mails diretamente para a Internet. Em qualquer um dos casos, máquinas não servidoras podem compartilhar a mesma configuração para **sendmail**. Talvez você queira distribuir a configuração com uma ferramenta como **rdist** ou **rsync**.[9]

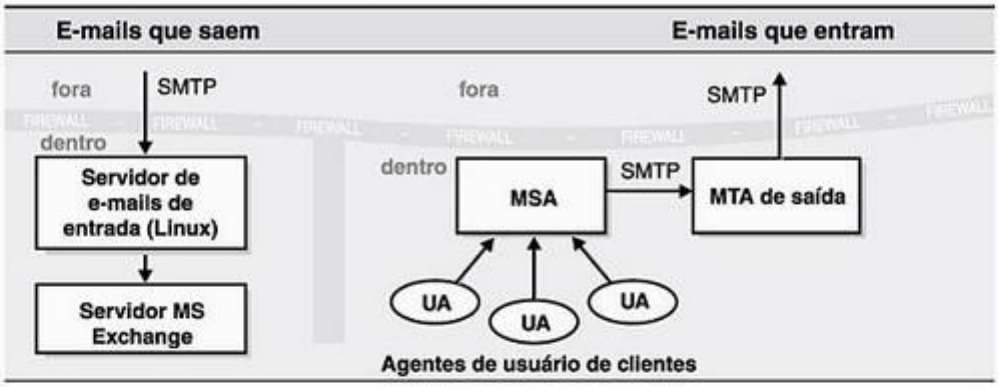
Sites que usam software como o Microsoft Exchange e Lotus Notes, mas que não se sentem à vontade para expor diretamente estes aplicativos para a Internet, poderão utilizar um projeto que se baseie no modelo apresentado no Diagrama D.

Independentemente de o projeto escolhido, certifique-se de que sua configuração **sendmail**, seus registros MX do DNS e suas regras de firewall estejam implementando a mesma política em relação ao e-mail.

9. **rsync** é incluso com o Red Hat e o SuSE e é apenas um **apt-get** no Debian.

Snow666

Diagrama D Arquitetura de sistema de e-mail - 2º diagrama



Utilizando centrais de e-mail (mail homes)

É conveniente para os usuários receberem e preservarem suas mensagens de e-mail em uma única máquina, mesmo se eles queiram acessar estas mensagens de vários sistemas diferentes. Podemos implementar centrais de e-mail usando o arquivo **aliases** ou um banco de dados LDAP (veja a seção 19.4). Podemos fornecer acesso remoto à caixa postal de cada usuário via IMAP ou POP.

O esquema de nomes alternativos que utilizamos permite que os arquivos de nomes alternativos sejam os mesmos em todas as máquinas dentro de um domínio administrativo. Essa uniformidade é uma grande conquista sob o ponto de vista do administrador. (Partimos do pressuposto que os nomes de login e os UIDs sejam exclusivos ao longo de todas as máquinas, uma política que recomendamos enfaticamente.)

Alguns sites centralizam e-mails exportando **/var/spool/mail** via NFS. Problemas de bloqueio com arquivos NFS podem fazer com que os usuários percam e-mails ou tenham seus arquivos de spool corrompidos. Culpar o NFS, o **sendmail** ou os agentes de entrega em relação a bloqueio apropriado não ajuda o pobre usuário cuja caixa postal foi corrompida (porém, **sendmail** é sempre inocente, já que ele na verdade jamais entrega e-mails).

Algumas implementações do NFS (tais como aquelas em servidores NFS dedicados) incluem um gerenciador de bloqueio que funciona perfeitamente. A maioria das implementações não suporta bloqueio ou o faz de forma inadequada. Alguns sites simplesmente ignoram o problema de bloqueio e torcem pelo melhor e outros exigem que os usuários leiam e-mails no servidor de arquivos.

Nosso conselho é não usar um **/var/spool/mail** compartilhado por NFS. O LMTP (Local Mail Transport Protocol, protocolo de transporte de e-mails local) é uma boa alternativa para o NFS para comunicação com um servidor de arquivos que armazena e-mails.

Utilizando IMAP ou POP

IMAP e POP são protocolos que fazem o download de e-mails para a máquina desktop de um usuário quando ela estabelece conexão com a rede. É a maneira ideal de gerenciar e-mails, especialmente para hosts que nem sempre estão conectados, seja devido a estarem desligados quando não estão em uso ou por estarem num ambiente doméstico compartilhando a linha telefônica com os adolescentes da casa. É exigida uma senha; certifique-se de usar uma versão que não transmita essa senha em texto puro através da Internet.

Gostamos do IMAP (Internet Message Access Protocol, protocolo de acesso a mensagens Internet) mais do que o POP. Ele entrega sua correspondência uma mensagem por vez e não todas de uma só vez, que é muito mais gentil com a rede (especialmente num link lento) e melhor para alguém que está sempre se deslocando de um lugar para outro. Ele é especialmente interessante para lidar com os anexos gigantes que alguns coleguinhas adoram enviar: você pode percorrer os cabeçalhos de suas mensagens de e-mail e não fazer o download dos anexos até estar preparado para fazê-lo. O IMAP gerencia pastas de e-mail entre vários sites, por exemplo, entre o seu servidor de e-mails e o seu PC. E-mails que permanecem no servidor Linux podem fazer parte do cronograma normal de backup. www.imap.org contém um bocado de informações sobre o IMAP e uma lista de implementações disponíveis.

O POP (Post Office Protocol, protocolo de correio) é similar, porém pressupõe um modelo no qual é feito o download de todos os e-mails do servidor para o PC. Ele pode ser eliminado do servidor (em cujo caso ele poderá ficar sem backup) ou então salvo no servidor (em cujo caso seu arquivo spool para e-mail vai crescendo cada vez mais). O paradigma "toda a caixa postal de uma só vez" é difícil na rede e menos flexível para o usuário. Pode ser realmente

www.snow666.blogspot.com
Snow666

lento em linhas discadas se você for um destes colecionadores de pequenos itens inúteis e tiver um arquivo de spool de e-mail grande. Os e-mails acabam ficando espalhados com o POP.

Ambos os protocolos podem se tornar devoradores de recursos caso os usuários jamais eliminem qualquer mensagem. No caso do IMAP, leva um tempão carregar os cabeçalhos de todas as mensagens de e-mail; o POP transmite toda a caixa postal. Certifique-se de que seus usuários compreendam a importância de eliminar mensagens ou de armazená-las em pastas locais.

Uma implementação razoável de POP3, a versão atual deste protocolo, se encontra disponível no site da Qualcomm em www.eudora.com/qpopper. **qpopper** é um servidor POP que inclui autenticação TLS/SSL entre o servidor e o cliente e mensagens criptografadas. Há vários outros servidores e clientes POP para Linux em www.ibiblio.org/pub/Linux/system/mail/pop. Certifique-se de escolher um que esteja sendo mantido ativamente.

O software de servidor IMAP se encontra disponível em www.washington.edu/imap. Não é necessária nenhuma configuração, exceto colocar as entradas IMAP apropriadas nos arquivos **/etc/services** e **/etc/inetd.conf** e certificar-se de que seu firewall (se realmente houver um) não impeça o IMAP de funcionar. O IMAP foi acusado, no passado, de problemas relativos à segurança; consulte os alertas CERT e certifique-se de obter a versão mais recente do IMAP, especialmente se houver boletins de segurança com datas posteriores à da sua distribuição Linux. A Carnegie Mellon University desenvolveu um servidor IMAP chamado Cyrus IMAP que suporta o protocolo POP também. Preferimos essa do que a implementação UW (Universidade de Washington) IMAP.

Todas as três distribuições Linux usadas como exemplo neste livro incluem um servidor IMAP chamado **imapd** e um cliente, **fetchmail**, que fala tanto IMAP quanto POP. O **imapd** do Red Hat é o servidor Cyrus IMAP da CMU; o SuSE e o Debian usam o IMAP da Universidade de Washington. O Red Hat também inclui o **pop3d**, um servidor POP. Para não ser ultrapassado, o SuSE inclui três: **qpopper** (que o SuSE rebatizou de **popper**), **pop2d** e **pop3d**. O Debian possui várias ferramentas para gerenciamento de caixas postais com IMAP; **man -k imap** lhe informará os nomes delas. **apt-cache search imap** dará uma saída prolixa, mas com mais informações sobre ferramentas para IMAP/POP.

19.4 Nomes Alternativos de E-Mail

Nomes alternativos permitem que e-mails sejam redirecionados pelo administrador de sistemas ou por usuários individuais.[10] Eles podem definir listas de discussão, encaminhar e-mails entre as máquinas ou permitir que os usuários sejam referidos por mais de um nome. O processamento de nomes alternativos é recursivo, de modo que é permitido para um nome alternativo apontar para outros destinos que são nomes alternativos em si.

sendmail suporta vários mecanismos de nomes alternativos: LDAP (Lightweight Directory Access Protocol), NIS e NIS+ da Sun, NetInfo da NeXT/Apple, bancos de dados de direcionamento de e-mails e vários arquivos de nomes alternativos que os usuários e os administradores de sistemas podem configurar.

Veja mais informações sobre o LDAP ainda nesta seção.

Caso queira usar o conceito de centrais de e-mail e tenha um site grande e complexo, recomendamos que você implemente centrais de e-mail armazenando seus nomes alternativos num servidor LDAP. As outras alternativas possuem várias desvantagens. Os arquivos de nomes alternativos planos têm de ser mantidos pelo root. A representação em forma de bancos de dados destes têm de ser reconstruída a cada mudança e eles precisam ser replicados em cada máquina que entrega e-mails. Para um site pequeno ou simples é melhor simplesmente usar o arquivo plano **/etc/mail/aliases** e o comando **newaliases**. Alguns sites que possuem muitos domínios virtuais mantêm fragmentos de arquivos de nomes alternativos distintos para cada domínio e os mesclam através de um script para formar o verdadeiro arquivo de nome alternativo.

O LDAP não funciona diretamente para nomes alternativos de listas de discussão. Entretanto, ele pode ser usado para retornar nomes alternativos que apontam para programas como o **majordomo** ou para arquivos `:include:`. Esses recursos permitem que o LDAP funcione bem em ambientes que tenham de oferecer suporte a listas de discussão.

sendmail usa um servidor LDAP de maneira muito parecida com a do servidor de nomes DNS. Ele chama o servidor DNS para converter nomes em endereços IP para que possam ser enviadas mensagens. Ele chama o servidor LDAP para pesquisar nomes alternativos para redirecionar as mensagens para o lugar correto. Em ambos os casos, as pesquisas passaram de arquivos planos (**/etc/hosts** e **/etc/aliases**) para bancos de dados, com os servidores gerenciando as consultas.

10. Tecnicamente falando, os nomes alternativos são configurdos somente por administradores de sistemas. O controle do usuário sobre o direcionamento de e-mails por meio de um arquivo **.forward** não é realmente um nome alternativo, mas nós os agrupamos aqui.

Snow666

Abordamos o LDAP de forma mais detalhada em três locais deste capítulo e também o mencionamos no Capítulo 18, *Compartilhando Arquivos de Sistema*. Introduzimos e explicamos o LDAP um pouco mais à frente neste capítulo. A seguir descrevemos a interação do LDAP com **sendmail** e com o arquivo de configuração de **sendmail** (seção 19. 8). Nosso primeiro exemplo usa LDAP para implementar nomes alternativos e hospedagem virtual.

Porém, antes de mergulharmos no LDAP, descreveremos primeiramente os mecanismos de nomes alternativos tradicionais. Os nomes alternativos podem ser definidos nos três locais seguintes (infelizmente, com três sintaxes diversas):

- No arquivo de configuração de um agente de usuário (pelo usuário remetente)
- No arquivo global **/etc/mail/aliases** (pelo administrador de sistemas)
- No arquivo de encaminhamento de um usuário, **~/.forward** (pelo usuário receptor)[11]

O agente de usuário procura nomes alternativos nos arquivos de configuração do usuário e os expande antes de introduzir a mensagem no sistema de e-mail. O agente de transporte, **sendmail**, procura nomes alternativos no arquivo global **aliases** e depois nos arquivos de encaminhamento do receptor. Os nomes alternativos são aplicados somente a mensagens que **sendmail** considera serem locais.[12]

Eis alguns exemplos de nomes alternativos no formato de arquivo **aliases**:

```
nemeth: evi
evi: evi@mailhub
authors: evi,garth,trent
```

A primeira linha diz que o e-mail enviado a "nemeth" deve ser entregue ao usuário evi na máquina local. A segunda linha diz que todos os e-mails para evi devem ser entregues na máquina mailhub e a terceira linha que o e-mail endereçado aos autores ("authors") deve ser entregue aos usuários evi, garth e trent. A recursão é suportada, de modo que o e-mail enviado a nemeth na verdade acaba indo para evi@mailhub.

Veja o Capítulo 18 para obter maiores informações sobre o NIS.

O caminho para o arquivo global **aliases** é especificado no arquivo de configuração de **sendmail** — ele é **/etc/aliases** tanto no Red Hat quanto no SuSE e no Debian. **/etc/mail/aliases** é na verdade o local "padrão". Os sites podem ter vários arquivos **aliases** e eles também podem usar maneiras alternativas para armazenar associações de nomes alternativos como NIS ou arquivos de bancos de dados.

O formato de uma entrada no arquivo **aliases** é

```
nome-local: receptor1,receptor2,_
```

onde *nome-local* é o endereço original a ser comparado em relação às mensagens que chegam à lista de receptores contém endereços de receptores ou outros nomes alternativos. As linhas recuadas são consideradas continuações das linhas precedentes.

Sob o ponto de vista do e-mail, o arquivo **aliases** prevalece sobre **/etc/passwd**, de modo que a entrada

```
david: david@somewhere-else.edu
```

impediria que o usuário local david recebesse qualquer e-mail. Portanto, os administradores e as ferramentas **adduser** devem verificar tanto o arquivo **passwd** quanto o arquivo **aliases** ao selecionarem novos nomes de usuários.

O arquivo **/etc/mail/aliases** sempre deve conter um nome alternativo chamado "postmaster" que encaminha e-mails para aquele que faz a manutenção do sistema. Um nome alternativo para mensagens automáticas de **sendmail** também tem de estar presente; este normalmente é chamado de Mailer-Daemon e freqüentemente lhe é aplicado o nome alternativo postmaster.

Você deve redirecionar os e-mails do root para os administradores de seu site ou para alguém que entre no sistema todos os dias. As contas bin, sys, daemon, nobody e hostmaster (e qualquer outra conta de pseudo-usuário que você configurar) também devem ter nomes alternativos que encaminhem e-mails para um ser humano. O arquivo **sendmail/aliases** da distribuição é um bom modelo para os nomes alternativos globais que devem ser inclusos. Ele também inclui sugestões relativas à segurança e um exemplo de como algumas solicitações de usuário comuns são direcionadas na Berkeley.

---

11. **~/.forward** é o local-padrão que **sendmail** procura. Porém, podemos utilizar outro caminho configurando a opção `ForwardPath` de **sendmail**.
12. De fato, isso não é totalmente verdadeiro. Se adicionarmos o flag `F=A` ao mailer SMTP, também poderemos implementar nomes alternativos remotos.

Snow666

sendmail detecta loops que poderiam fazer com que o e-mail ficasse indo e voltando para sempre contando o número de linhas Received no cabeçalho de uma mensagem e retornando-a ao remetente quando a contagem atingisse um limite preestabelecido (normalmente 25).[13] Cada visita a uma nova máquina é chamada de "salto" no jargão do **sendmail**; retornar uma mensagem ao remetente é conhecido como "bouncing" ("ricochete"). A sentença anterior, devidamente transformada em jargão, ficaria: "A mensagem de e-mail ricocheteia após 25 saltos".[14]

Além de uma lista de usuários, os nomes alternativos podem se referir a:

- Um arquivo contendo uma lista de endereços
- Um arquivo para o qual as mensagens devem ser anexadas
- Um comando para o qual as mensagens devem ser dadas como entrada

Já que o remetente de uma mensagem determina totalmente seu conteúdo, esses destinos de entrega eram freqüentemente aproveitados por hackers. **sendmail** se tornou muito exigente em relação à propriedade e às permissões de tais arquivos e comandos. Para suplantar a paranóia de **sendmail**, temos de configurar uma das opções `DontBlameSendmail`, assim chamadas para desencorajá-lo a fazê-lo. Infelizmente, as mensagens de erro que **sendmail** produz ao encontrar permissões ou propriedades inseguras nem sempre são claras.

Obtendo listas de discussão de arquivos

A diretiva `:include:` é um excelente meio para permitir que os usuários gerenciem suas próprias listas de discussão. Ela permite que os membros de um nome alternativo sejam extraídos de um arquivo externo em vez de serem listados diretamente no arquivo **aliases**. O arquivo também pode ser modificado localmente sem a intervenção do administrador de sistemas responsável pelo arquivo global **aliases**.

Ao configurar a lista, o administrador tem de introduzir o nome alternativo no arquivo global **aliases**, criar o arquivo **include** e aplicar **chown** neste arquivo **include** para o usuário que mantém a lista de discussão. Por exemplo, o arquivo **aliases** poderia conter

```
sabook: :include:/usr/local/mail/usah.readers
```

O arquivo **usah.readers** deve estar num sistema de arquivos local, não num sistema de arquivos montado com NFS[15], e deve ser gravável somente pelo próprio dono. Para ser realmente completo, também devemos incluir nomes alternativos para o proprietário da lista de discussão para que erros (ricochetes) sejam enviados ao proprietário da lista, e não ao remetente de uma mensagem endereçada à lista:

```
owner-sabook: evi
```

Veja a seção *Listas de discussão e software de discussão*, mais à frente, para obter maiores detalhes sobre listas de discussão e a interação destas com o arquivo **aliases**.

Enviando e-mails a arquivos

Se o destino de um nome alternativo for um nome de caminho absoluto (entre aspas duplas se ele incluir caracteres especiais), as mensagens serão anexadas ao arquivo especificado. O arquivo já tem de existir.

Por exemplo:

```
complaints: /dev/null
```

É útil ser capaz de enviar e-mails a arquivos e programas, mas este recurso introduz preocupações relativas à segurança e, portanto, é restrito. Essa sintaxe é válida somente no arquivo **aliases** e no arquivo **.forward** de um usuário (ou, num arquivo que é interpolado em um destes arquivos via `:include:`). Um nome de arquivo não é entendido como um endereço normal, de forma que e-mails endereçados a /etc/passwd@host.domínio ricochetearão.

---

13. O limite-padrão de saltos é de 25, contudo você pode modificá-lo no arquivo de configuração.
14. Não fomos consistenets na terminologia deste capítulo, algumas vezes chamando uma mensagem retornada de "ricochete" e algumas vezes chamando-a de "erro". O que realmente quisemos dizer foi que o DSN (Delivery Status Notification, notificação de estado de entrega) foi gerado. Tal notificação normalmente significa que uma mensagem não pôde ser entregue e que, portanto, ela retornará ao remetente.
15. Se o sistema de arquivos NFS for montado com a opção "hard" e o NFS falhar, **sendmail** irá bloquear com vários manipuladores de arquivos abertos e vários processos em estado de espera. Talvez você possa ficar sem IDs de processos ou manipuladores de arquivos e ter de reinicializar a máquina para corrigir as coisas.

Snow666

Alguns agentes de usuário permitem que você salve mensagens de e-mail em um arquivo local (como se fosse uma pasta de caixa de saída). Entretanto, essa cópia da mensagem é salva pelo agente de usuário e jamais é realmente processada pelo sistema de e-mail.

Se o arquivo de destino for referido no arquivo **aliases**, ele terá de permitir a gravação para todo o mundo (o que não é recomendado), terá de ser setuid mas não executável, ou o seu proprietário terá de ser o usuário-padrão de **sendmail**. A identidade do usuário-padrão é configurada com a opção `DefaultUser`. Normalmente ela é mailnull, sendmail, daemon ou UID 1, GID 1.

Se o arquivo for referido num arquivo **.forward**, ele deve pertencer e ser gravável pelo receptor da mensagem original, que deve ser um usuário válido com uma entrada no arquivo **/etc/passwd** e um shell válido que é listado em **/etc/shells**. Para arquivos pertencentes à root, use modo 4644 ou 4600, setuid, mas não executável.

Enviando e-mails para programas

Um nome alternativo também pode direcionar e-mails para a entrada-padrão de um programa. Esse comportamento é especificado com uma linha como

```
autoftp: "|/usr/local/bin/ftpserver"
```

É até mais fácil criar brechas de segurança com esse recurso do que enviando um e-mail a um arquivo, de modo que mais uma vez ele é permitido somente em arquivos **aliases**, **.forward** ou `:include:`. No arquivo **aliases**, o programa executa como usuário-padrão do **sendmail**; caso contrário, o programa executa como proprietário do arquivo **.forward** ou `:include:`. Esse usuário tem de ser listado no arquivo **/etc/passwd** com um shell válido (em **/etc/shells**).

O programa entregador local de e-mails (mailer) muda seu diretório de trabalho para o diretório de origem do usuário (ou, se este diretório for inacessível, para o diretório-raiz) antes de executar o comando que deve receber o e-mail. O padrão costumava ser o diretório de filas do **sendmail**, porém alguns shells baseados em **csh** objetaram.

Enviar e-mails para programas é uma grande brecha de segurança potencial. Para maior segurança, não use **/bin/sh** como o shell do seu programa entregador local de e-mail; use, em vez disso, o shell restrito de **sendmail**, **smrsh**. Veja a seção 19.11 para obter maiores informações sobre **smrsh**.

Exemplos de nomes alternativos

Eis alguns nomes alternativos típicos que um administrador de sistemas poderia usar.

```
# Nomes alternativos exigidos[16]

postmaster: trouble, evi
postmistress: postmaster
MAILER-DAEMON: postmaster
hostmaster: trent
abuse: postmaster
webmaster: trouble, trent
root: trouble, trent

# include para nome alternativo de problemas local
trouble: :include:/usr/local/mail/trouble.alias
troubletrap: "/usr/local/mail/logs/troublemail"
tmr: troubletrap,:include:/usr/local/mail/tmr.alias

# conveniências de sysadmin
diary: "/usr/local/admin/diary"
info: "|/usr/local/bin/sendinfo"

# nomes alternativos de classes que mudam a cada semestre
sa-class: real-sa-class@nag
real-sa-class: :include:/usr/local/adm/sa-class.list
```

Nesse exemplo, gostaríamos que usuários de todo o campus fossem capazes de enviar e-mails para um único nome alternativo para problemas ("trouble") sempre que ocorresse um problema. Relatórios de problemas sempre

16. Uma mentirinha. Somente postmaster e MAILER-DAEMON são realmente exigidos (pelas RFCs), porém é convencional incluir hostmaster, abuse e webmaster também.

www.snow666.blogspot.com
Snow666

devem ser direcionados a um grupo de administradores de redes locais apropriado. Particularmente, gostaríamos de configurar nomes alternativos de e-mails para que

- E-mails referentes a problemas sempre fossem destinados a um grupo apropriado.
- Uma única versão do arquivo de nomes alternativos fosse usada em todos os hosts.
- Grupos de administradores individuais controlassem suas próprias listas de distribuição.
- Uma cópia de todos os e-mails referentes a problemas fosse para um arquivo de log local para cada grupo.

A configuração anterior satisfaz esses objetivos extraindo a definição do nome alternativo para problemas de um arquivo contido em cada máquina. E-mails enviados para os endereços trouble@anchor e trouble@boulder acabariam indo para lugares diferentes muito embora anchor e boulder usem o mesmo arquivo **/etc/mail/aliases**.

E-mails referentes a problemas são normalmente manipulados numa determinada máquina em cada local. Por exemplo, o arquivo **trouble.alias** em uma máquina escrava poderia conter o endereço

```
trouble@mestre
```

para que e-mails referentes a problemas fossem para a máquina mestra apropriada.

Quando uma mensagem referente a problemas é determinada, ela é enviada para o nome alternativo tmr, que significa "trouble mail readers", ou seja, "leitores de mensagens referentes a problemas". O nome alternativo tmr arquiva a mensagem para o nome alternativo de captura de problemas e também a envia a uma lista de usuários extraídos de um arquivo na máquina mestra. Acrescentar administradores novatos à lista tmr é uma excelente maneira de fazê-los observar as questões de suporte que surgem, as respostas dos administradores e o tom bajulador apropriado que deve ser usado com os usuários (isto é, clientes).

Esse mecanismo é apenas a ponta do iceberg de um sistema completo de acompanhamento de passagem de problemas chamado **queuemh**, que se baseia no agente de usuário **mh**.

O nome alternativo sa-class possui dois níveis, de modo que o arquivo de dados contendo a lista de estudantes somente precisa ser mantido numa única máquina, nag. O exemplo de nome alternativo sabook, apresentado anteriormente, deveria realmente ter este mesmo tipo de indireção para que o arquivo não precisasse ser replicado. O nome alternativo diary é conveniente e funciona bem como uma técnica de extração de documentação para administradores de sistemas novatos que ficam indignados em documentar o que eles fazem.

Os administradores de sistemas podem criar memoriais de eventos importantes na vida da máquina (atualizações de sistemas operacionais, mudanças de hardware, quebras etc.) enviando e-mail para o arquivo diary. Não coloque o arquivo num sistema que contenha seus arquivos de logs; isso permitira que hackers preenchessem o sistema de arquivos e impedissem que syslog gravasse suas entradas de log (e, portanto, encobrindo seus rastros).

Encaminhamento de e-mails

aliases é um arquivo de configuração global que deve ser mantido por um administrador. Se os usuários quiserem redirecionar seus próprios e-mails (e o seu site não usar POP ou IMAP para acessar e-mail), eles podem fazê-lo criando arquivos **.forward** em seus diretórios de origem. **sendmail** sempre pesquisa o diretório de origem de um usuário em busca de um arquivo **.forward** a menos que a variável `ForwardPath` esteja ativada e altere a localização-padrão. É conveniente usar um arquivo **.forward** quando um usuário quer receber e-mails num determinado host ou quando alguém deixa o seu site e quer que o e-mail seja encaminhado para uma nova localização.

Um arquivo **.forward** é formado por uma lista de endereços separados por vírgulas numa única linha ou várias entradas em linhas distintas. Por exemplo,

```
evi@ipn.caida.org
evi@xor.com
```

ou

```
\mcbryan, "/home/mcbryan/archive", mcbryan@flsupil.gmd.de
```

No primeiro exemplo, e-mails para evi não são entregues na máquina local, mas sim encaminhados à máquina ipn no CAIDA em San Diego e para xor.com. A segunda entrada é de um usuário que não confia nos sistemas de e-mail e quer que seus e-mails sejam replicados em três locais: o spool de e-mails regular na máquina, um arquivo permanente de todos os e-mails que chegam e um endereço temporário na Alemanha por onde ele está viajando no momento. A barra invertida antes de seu nome de usuário diz para entregar e-mails localmente independentemente do que os arquivos de nomes alternativos ou de encaminhamento possam instruir.

Snow666

Para mudanças temporárias no direcionamento de e-mails, o emprego de um arquivo **.forward** é preferível em vez do arquivo global **aliases**. O custo adicional (tempo de processamento e tempo das pessoas) necessário para alterar os nomes alternativos globais é muito elevado.

O arquivo **.forward** de um usuário tem de pertencer ao usuário e não pode ser gravável pelo grupo ou por todo o mundo. Se **sendmail** acredita que o caminho de diretórios para o arquivo **.forward** seja seguro (isto é, as permissões a partir da raiz para baixo estão OK), ele pode ser um link; caso contrário, ele não poderá ser um link. **sendmail** ignora arquivos de encaminhamento nos quais as permissões pareçam suspeitas; as permissões no diretório-pai também têm de ser seguras (graváveis somente pelo usuário ao qual os arquivos pertencem).

Naturalmente, **sendmail** tem de ser capaz de acessar o diretório de origem de um usuário na máquina onde os e-mails são entregues para determinar se ele contém um arquivo **.forward**. Mudanças permanentes de endereço devem ser colocadas no arquivo **/etc/mail/aliases,** pois os arquivos e o diretório de origem de um usuário podem, eventualmente, serem removidos.

sendmail possui um recurso muito interessante, `FEATURE(`redirect')`, que ajuda no gerenciamento de mudanças de endereços de e-mail permanentes. Se um nome alternativo aponta para usuario@novosite.REDIRECT, os e-mails serão retornados ao remetente com uma notificação do novo endereço. A mensagem não é encaminhada para o novo endereço, de modo que o remetente terá de atualizar a sua lista de endereços e reenviar a mensagem.

Você poderá configurar **sendmail** para suportar um diretório central para arquivos **.forward**, porém os usuários não esperam esta configuração. A localização dos arquivos **.forward** é controlada pela opção `ForwardPath`, que normalmente aponta para este diretório central e depois para o diretório de origem do usuário. O arquivo de domínio **generic.m4** ilustrado na seção 19.7 contém um exemplo de um local centralizado para arquivos **.forward**. Uma entrada no arquivo **aliases** global tem precedência sobre uma entrada num arquivo **.forward**.

Já que esses arquivos são mantidos por pessoas diferentes, os usuários têm de ser cuidadosos para inadvertidamente não criarem loops de e-mails. Se um usuário de uma rede tiver uma central de e-mail (e, conseqüentemente, uma entrada no arquivo global **aliases**), esse usuário não poderá usar um arquivo **.forward** para redirecionar e-mails para uma outra máquina que compartilhe os mesmos endereços alternativos. Por exemplo, na Universidade do Colorado, onde usamos um arquivo **aliases** que abrange todo o site, uma entrada como

```
evi: evi@boulder
```

e um arquivo **.forward** na máquina boulder contendo

```
evi@anchor.cs
```

criaria um loop. E-mails endereçados a evi seriam encaminhados a boulder, onde o arquivo **.forward** faria com que eles fossem enviados para anchor no subdomínio cs. O arquivo **aliases** em anchor faria com que eles fossem encaminhados de volta para boulder e assim por diante. Após 25 saltos, os e-mails seriam retornados ao remetente.

Notificar um usuário sobre loop de e-mails é desafiador se seu modo primário de comunicação for e-mail. E-mails para *\usuario*[17] entrega a mensagem na máquina local, independentemente do que o arquivo **aliases** global ou o arquivo **.forward** do usuário possa instruir. Se a máquina local é onde o usuário espera ler os e-mails, tudo bem; caso contrário, envie um e-mail ao postmaster para relatar o loop ou então pegue o telefone!

O banco de dados de nomes alternativos hashed

Já que as entradas no arquivo **aliases** não se encontram em nenhuma ordem em particular, seria ineficiente para **sendmail** procurar este arquivo diretamente. Em vez disso, uma versão é construída com o sistema de banco de dados Berkeley DB ou com o sistema de banco de dados **ndbm** que é padrão na maioria das versões do UNIX. Esse hashing aumenta significativamente a velocidade de pesquisas de nomes alternativos, especialmente quando o arquivo se torna grande.

Os arquivos derivados de **/etc/mail/aliases** se chamam **aliases.db** no DB e **aliases.dir** e **aliases.pag** no **ndbm**. O arquiv **dir** é um índice para o arquivo **pag**, no qual os dados efetivamente residem. Cada vez que você mudar o arquivo **aliases**, terá de reconstruir o banco de dados hashed com o comando **newaliases**. **newaliases** é realmente apenas **sendmail** em forma dissimulada com flags de linha de comando (**-bi**) que o informam para reconstruir o banco de dados. Salve a saída de erro gerada se você executar **newaliases** automaticamente — talvez você tenha introduzido erros de formatação.

---

17. Tavlez você tenha de usar duas ou mais barras invertidas para que uma delas passe o shell e vá para **sendmail**.

Snow666

Veja o Capítulo 18 para obter maiores informações sobre o NIS.

Ao compilar **sendmail**, você deve incluir suporte à biblioteca de banco de dados para as rotinas **dbm/ndbm**, as novas rotinas Berkeley DB ou ambas. Se ambas forem inclusas e **sendmail** precisar criar um arquivo de banco de dados mas o tipo não for especificado, **sendmail** usará DB. Se estiver usando NIS, **sendmail** *cria* ambos, porém *usa* a versão DB caso o caminho para o arquivo **aliases** inclua **/yp/**.

A biblioteca DB, escrita e mantida por Keith Bostic e Margo Seltzer, se encontra disponível em www.sleepycat.com. Ela é muito melhor (acesso mais rápido, arquivos de bancos de dados menores) do que o sistema **ndbm**. DB é código-fonte aberto e gratuito a não ser que você o inclua num produto proprietário; neste caso, você precisará de uma licença.

Listas de discussão e software de discussão

Uma lista de discussão é um nome alternativo gigante que envia uma cópia de cada mensagem postada nela a cada pessoa que se afiliou à lista. É semelhante a um grupo de notícias Usenet que é entregue via e-mail. Algumas listas de discussão possuem milhares de receptores.

Listas de discussão são normalmente especificadas no arquivo **aliases,** porém mantidas em um arquivo externo. Algumas convenções de nomes padrão são compreendidas por **sendmail** e pela maioria dos programas para lista de discussão. Usuários experientes vieram a depender delas também. As mais comuns são o sufixo "-request" e o prefixo "owner-", usados para atingir os mantenedores da lista. As convenções são ilustradas pelos seguintes nomes alternativos:

```
mylist: :include:/etc/mail/include/mylist
owner-mylist: mylist-request
mylist-request: evi
owner-owner: postmaster
```

Nesse exemplo, mylist é o nome da lista de discussão (mailing list). Os membros são lidos no arquivo **/etc/mail/include/mylist**. Ricochetes do mailing para a lista são enviados ao seu proprietário, evi, assim como são as solicitações para se juntar à lista. A indireção de "owner" para "request" para evi é útil, pois o endereço do proprietário (neste caso, mylist-request) se torna o endereço Return-Path em cada mensagem enviada à lista. mylist-request é um pouco melhor do que o real mantenedor para este campo. Erros em mensagens para o nome alternativo owner-mylist (na verdade, evi) seriam enviados para owner-owner.

Existem outras convenções; por exemplo, sendmail.com usaria:

```
mylist: :include:/etc/mail/include/mylist
owner-mylist: mylist-owner
mylist-owner: evi
mylist-request: evi
```

O caso no qual uma mensagem não pode ser entregue é chamado de ricochete. O caso no qual a mensagem de erro enviada sobre o ricochete não puder ser entregue é chamado de ricochete duplo. Portanto, em nosso exemplo, ricochetes duplos são enviados para owner-owner ou para postmaster.

Se você usa um arquivo de nomes alternativos válido para todo o site, poderá precisar acrescentar um nível extra de indireção apontando mylist para myreallist@master para que o arquivo de dados contendo a lista de membros precise existir somente num único local.

Vários pacotes de software automatizam a manutenção de listas de discussão. Eles normalmente permitem que os usuários se acrescentem e se eliminem da lista, obtenham informações sobre a lista e obtenham arquivos via e-mail. Alguns dos gerenciadores de listas de discussão populares (e seus locais para download) são

- Majordomo, em www.greatcircle.com (incluso no SuSE)
- Mailman, o processador GNU de listas de discussão, em www.list.org
- ListProc, em www.cren.net
- SmartList, em ftp.informatik.rwth-aachen.de
- **listmanager** em www.listmanager.org
- LISTSERV Lite, em www.lsoft.com (versão gratuita do produto comercial LISTSERV)

Uma excelente FAQ de Norm Aleks sobre software para lista de discussão se encontra disponível em rtfm.mit.edu via FTP (procure em comp.answers/mail/list-admin). Infelizmente, a FAQ não é mais mantida e, por-

Snow666

tanto, pode estar um tanto desatualizada. Porém, ela ainda inclui informações úteis para ajudá-lo a escolher os melhores programas para listas de discussão de acordo com suas necessidades.

Geralmente, SmartList é pequeno e simples, ListProc é grande e complexo e os demais são intermediários. Eles diferem em suas filosofias de manutenção de listas, com alguns tendendo para sysadmins como administradores (ListProc) e outros tendendo para usuários como mantenedores (Majordomo, Mailman, SmartList, LISTSERV Lite). Majordomo, Mailman, **listmanager** e LISTSERV Lite suportam administração remota; o mantenedor da lista não precisa nem mesmo ter um login na máquina onde a lista se localiza, pois todas as transações ocorrem via e-mail. A maioria dos pacotes para listas de discussão permite que as informações postadas à lista sejam montadas em resenhas, algumas de forma automática (ListProc, Mailman, **listmanager** e LISTSERV Lite) e outras via configuração manual (SmartList e Majordomo).

Mailman é o nosso gerenciador de listas favorito. É uma alegria administrar e permitir que mantenedores de listas ajustem todos os recursos de suas próprias listas. ListProc e LISTSERV Lite são proprietários: o primeiro é muito caro, o outro somente binário e manco. Não experimentamos SmartList, mas gostamos de **procmail**, do qual ele depende.

Descrevemos brevemente cada um desses pacotes a seguir. Para maiores detalhes, veja a documentação que acompanha cada pacote ou o livro da O'Reilly, *Managing Mailing Lists* de Alan Schwartz e Paula Ferguson.

Majordomo

Majordomo é um pacote Perl/C disponível em www.greatcircle.com. Ele foi originalmente concebido por Brent Chapman. O desenvolvimento do Majordomo foi interrompido; Majordomo 2 foi totalmente reescrito mas ainda se encontra em teste beta, de modo que descrevemos somente o original. Entre as distribuições usadas como exemplo neste livro, somente o SuSE vem com o Majordomo. Apesar do que diz a página de manual (**/usr/lib/mail/majordomo**), ele se encontra oculto no diretório **/usr/lib/majordomo**.

Consulte a seção 19.3 para obter maiores informações sobre usuários confiáveis.

O Majordomo é executado como um usuário não privilegiado, tipicamente com o nome de usuário majordom ou mdom e grupo-padrão daemon. Já que o Linux suporta nomes de usuários longos (mais de 8 caracteres), você também pode usar majordomo como nome de login. O usuário tem de ser um que **sendmail** reconheça como "confiável" e, portanto, tem de ser mencionado em sua configuração **sendmail**, normalmente numa declaração `confTRUSTED_USERS`.[18]

Majordomo é configurado através do arquivo **majordomo.cf**, que é formado por comandos Perl válidos que inicializam variáveis, definem os diretórios nos quais as coisas se encontram (ou onde elas deveriam ser colocadas), especificam as listas a serem suportadas e configuram a manipulação de e-mail ricocheteado. Um programa auxiliar, **config-test**, testará o seu arquivo de configuração em busca de variáveis faltantes ou de sintaxe incorreta. O SuSE coloca o arquivo de configuração em **/etc** e deixa **config-test** com a distribuição Majordomo em **/usr/lib/majordomo**.

Majordomo requer que sejam instalados nomes alternativos especiais no arquivo **aliases** de **sendmail**. A maneira mais elegante de integrar esses nomes alternativos é criar um arquivo de nomes alternativos distinto usado somente para Majordomo (versões recentes de **sendmail** suportam vários arquivos de nomes alternativos). O arquivo contém um conjunto de nomes alternativos para o próprio Majordomo e um conjunto para cada lista de discussão que ele gerencia. A distribuição contém um modelo de arquivo de nomes alternativos, **majordomo.aliases**.

A pergunta mais comum feita pelos usuários sobre as listas de discussão é a seguinte: "Como faço para cancelar minha assinatura?". Para listas gerenciadas pelo Majordomo, a resposta para nome-da-lista@host é enviar um e-mail para o endereço majordomo@host com as palavras "unsubscribe nome-da-lista" ou "unsubscribe nome-da-lista endereço-de-email" no corpo da mensagem (e não na linha de assunto).

Com a primeira forma, você precisa enviar a mensagem de cancelamento de assinatura para o mesmo host que você usou ao assinar a lista; pela segunda forma, este host faz parte do endereço de e-mail. Veja a seção 19.2 para obter dicas sobre como coletar essas informações nos cabeçalhos de e-mail para poder cancelar a assinatura de maneira apropriada, mesmo que você tenha se esquecido de qual máquina usou ao se inscrever na lista. Algumas listas de discussão também aceitam e-mail para nome-da-lista-request@host apenas com a palavra "unsubscribe" no corpo.

Jamais, mas jamais mesmo, envie uma mensagem de cancelamento de assinatura para a própria lista. Se fizer isso, sua mensagem estará patenteando a todos os receptores da lista a sua incompetência.

---

18. Um usuário "confiável" tem permissão para alterar a linha de cabeçalho From de uma mensagem e reconstruir o arquivo **aliases**.

www.snow666.blogspot.com
Snow666

Mailman

Mailman, um acréscimo relativamente recente à frota de programas para listas de discussão, se encontra disponível em www.list.org ou nos arquivos GNU. Ele foi criado originalmente por John Viega e atualmente está sendo desenvolvido em colaboração com Ken Manheimer e Barry Warsaw. Assim como o Majordomo, Mailman é basicamente escrito numa linguagem de scripting com envoltórios em C, porém, neste caso a linguagem é o Python (disponível em www.python.org).

Mailman foi inspirado pelo emprego do Majordomo por parte dos autores e a frustração proveniente de erros-ricochete, configuração complicada dos recursos avançados como resenhas e listas moderadas e dificuldades de desempenho com grandes volumes de e-mail. Mailman dispõe de um script que importa listas Majordomo. Ele também possui alguma capacidade de detectar e controlar spam.

A grande alegação de Mailman para a fama é sua interface com a Web, que facilita o gerenciamento de uma lista por parte do moderador ou do do postmaster e também facilita os usuários em termos de assinatura, cancelamento de assinatura e de configuração de suas opções.

ListProc

ListProc é um velho conhecido dos programas para gerenciamento de listas de discussão. Ele foi escrito em 1991 por Anastasios Kotsikonas e mantido até por volta de 1994. Depois permaneceu parado por alguns anos, mas ressuscitou com uma nova versão em 1998. Ele costumava estar disponível gratuitamente no departamento de Ciências da Computação da Boston University, mas com algumas regras de licenciamento um tanto estranhas. Agora pode ser encontrado em www.cren.net por uma taxa de licenciamento pesada (US$ 2.000 por cópia, até mesmo para universidades). Esqueça-se de ListProc e adote um dos pacotes gratuitos e de fonte aberto.

SmartList

SmartList foi escrito originalmente por Stephen R. van den Berg, que também é o autor original do pacote **procmail**. Ele pode ser encontrado em ftp.informatik.rwth-aachen.de ou www.mindwell.com/smartlist. Este gerenciador usa **procmail**, de modo que você precisará fazer o download de **procmail.tar.gz** e de **SmartList.tar.gz** também. Provavelmente seja mais fácil simplesmente pegar o pacote Linux adequado ao seu sistema.

SmartList é compacto e simples. É uma combinação de código C, regras de **procmail** e scripts de shell. Ricochetes, a dor de cabeça das listas de discussão, são tratados de forma automática pelo software. Os usuários são eliminados automaticamente de uma lista após um certo número de ricochetes em relação a seus endereços. O SmartList requer uma entrada de login no arquivo **passwd** ("smart" ou talvez "list") que é um usuário confiável no arquivo de configuração de **sendmail**.

A instalação inclui **led**, um envoltório de bloqueio para edição que tenta proteger SmartList contra ser deixado com um arquivo de configuração incoerente e parcialmente editado.

listmanager

listmanager de Murray Kucherawy é escrito em C e, portanto, é mais rápido do que os pacotes construídos sobre uma linguagem de scripting como o Perl ou o Python. **listmanager** também melhora a velocidade usando o pacote de banco de dados DB de sleepycat.com em vez de arquivos planos e do sistema de arquivos. Sua lista de recursos ocupa aproximadamente uma página e inclui uma interface Web, resenhas e várias melhorias de segurança.

listmanager realmente se parece com um gerenciador de listas arrasador — o único ponto negativo é que o código-fonte ainda não está sendo distribuído. De acordo com a página Web em www.listmanager.org, o código está retido antes enquanto o autor finaliza um trabalho de limpeza para evitar possíveis embaraços. Há um binário Linux disponível.

LISTSERV Lite

LISTSERV Lite de Eric Thomas é uma versão gratuita de LISTSERV, o produto comercial da L-Soft International, Inc. Alguns dos recursos da versão comercial estão faltando nesta edição gratuita e o software está limitado a gerenciar 10 listas de discussão de até 500 pessoas cada. LISTSERV Lite precisa ser executado como o pseudo-usuário listserv, que deve ser o proprietário dos seus arquivos. Ele também gosta de ter um grupo listserv. LISTSERV Lite dispõe de uma interface Web tanto para assinar uma lista como para sua manutenção.

A distribuição se encontra disponível em www.lsoft.com. O código-fonte não é distribuído, mas binários pré-compilados e stubs para diversas versões do UNIX e do Linux são fornecidos. Se você estiver familiarizado com LISTSERV e tiver listas que o utilizem, talvez seja justificável o emprego de um gerenciador de listas enfraquecido e com somente binários. Caso esteja começando do zero, opte por uma das alternativas de fonte aberto e irrestrito mencionadas anteriormente.

Snow666

LDAP: Lightweight Directory Access Protocol

O LDAP é um protocolo que fornece acesso a um serviço de diretório administrativo genérico. Ele está no mercado há alguns anos, mas só recentemente começou a se tornar popular. Os administradores estão descobrindo que o LDAP é uma boa alternativa para várias coisas:

- Configuração de **sendmail**: nomes alternativos, domínios virtuais, central de e-mail, o banco de dados de acesso e tabelas
- Gerenciamento de usuários: nomes de login, senhas, hosts (por exemplo, Stanford University)
- Arquivos de configuração administrativos (por exemplo, SuSE Linux)
- Como substituto para o NIS
- Como um servidor de agendamento
- Para emprego com o PAM (Pluggable Authentication Modules. Módulos de autenticação plugáveis)

É previsto que o LDAP irá, eventualmente, se tornar um sistema de diretórios global usado para diversos tipos de propósitos. Infelizmente, ainda faltam ferramentas para automação de tarefas típicas desempenhadas pelos administradores de sistemas com LDAP.

O LDAP nasceu dos protocolos ISO e do serviço de diretórios X.500. Essa ascendência imediatamente sugere os termos complexo, inflado, prolixo, ruim etc., mas o L na sigla LDAP supostamente deve cuidar de tudo isto. As versões 1 e 2 deste protocolo foram padronizadas. A versão 3 está próxima disso. Felizmente, todas as versões são compatíveis com as anteriores. A versão 3 do protocolo LDAP suporta servidores hierárquicos; consultando um servidor para um bloco de dados pode resultar numa referência a um outro servidor. A versão 2 suportava dados hierárquicos, porém os servidores hierárquicos são suportados somente na versão 3.

Nomes alternativos de e-mail é um caso particularmente adequado para o LDAP, especialmente agora que **sendmail** suporta LDAP internamente. **sendmail** pode consultar o servidor LDAP para pesquisas em vez de fazê-las diretamente. O LDAP também é capaz de gerenciar direcionamento de e-mails e domínios virtuais. O suporte LDAP tem que ser compilado no binário **sendmail**.

Se você estiver procurando uma implementação LDAP, recomendamos o servidor produzido pelo grupo OpenLDAP em www.openldap.org. Esse grupo assumiu e aperfeiçoou o código de um servidor anterior e que foi desenvolvido na Universidade de Michigan. Para mais informações sobre software relacionado com LDAP, veja a seção18.5.

As entradas de bancos de dados LDAP relembram uma entrada termcap com nomes de variáveis mais longos. Os atributos (nomes de variáveis) no banco de dados LDAP ainda não estão totalmente padronizados e este fato pode resultar em incompatibilidades entre as diversas implementações.

Os atributos na primeira linha de uma entrada de banco de dados são definidos pelo arquivo de configuração do LDAP. Os exemplos desta seção supõem que o daemon de servidor LDAP (**slapd**, no caso do OpenLDAP) foi configurado com um nome root distinto (`rootdn`) de:

```
"cn=root, dc=synack, dc=net"
```

O atributo `dc` aparece duas vezes, pois os valores de componentes de domínios não podem conter um ponto; para expressar o domínio synack.net, duas entradas são necessárias. Os demais atributos, ou nomes de variáveis, podem ser o que você quiser. É indiferente o uso de letras maiúsculas ou minúsculas nos seus nomes. **sendmail** (cujo código procura nomes de atributos específicos e atribui a eles interpretações predeterminadas), o servidor LDAP e o construtor do banco de dados LDAP têm de cooperar e usar as mesmas convenções de atribuição de nomes.

Alguns atributos possíveis que podem aparecer na primeira linha de uma entrada de banco de dados (as chaves de banco de dados) são `dn` para um nome de domínio, `dc` para um componente de domínio, `o` para um nome de organização, `c` para um nome de país e `uid` para um ID exclusivo (por exemplo, um nome de login).

Como padrão, **sendmail** reconhece as seguintes tags de dados LDAP:

```
mailLocalAddress
mailRoutingAddress
mailHost
```

A versão 8.12 expande extensivamente esse conjunto-padrão (veja **cf/sendmail.schema**) e também permite que você defina o seu próprio esquema para usar quaisquer tags LDAP que deseje.

Eis um exemplo do arquivo **ldap.conf** da implementação OpenLDAP do **slapd**:

```
# Padrões para LDAP, arquivo ldap.conf, devem poder ser lidos por todos.
#
```

Snow666


```
BASE  dc=synack, dc=net
HOST  gw.synack.net
PORT  389
```

que suporta entradas de banco de dados da forma

```
dn: uid=jon, dc=synack, dc=net
objectClass: inetLocalMailRecipient
mailLocalAddress: jon@synack.net
mailRoutingAddress: stabilej@cs.colorado.edu
uid:jon
```

Quando `FEATURE(ldap_routing)` tiver sido especificado no arquivo de configuração **sendmail**, o receptor da mensagem é comparado com o campo `mailLocalAddress`. Se ele for coincidente, o e-mail será redirecionado para `mailRoutingAddress`. A linha `objectClass` tem de estar aí — ela provém da RFC preliminar que define a interação entre o LDAP e os sistemas de e-mail.

No host gw.synack.net, essa entrada de banco de dados corresponde ao nome alternativo:

```
jon: stabilej@cs.colorado.edu
```

Entediante, não é mesmo? As entradas de bancos de dados poderiam substituir as entradas típicas no arquivo **aliases** para definição de uma central de e-mail para cada usuário. Mas até que as ferramentas estejam amplamente disponíveis para adição, modificação e eliminação de entradas LDAP — isto é, ferramentas que não requeiram que você digite dezenas de vezes o equivalente a uma linha de nome alternativo — os benefícios de desempenho não justificam o tempo extra gasto pelos administradores de sistemas para manterem o banco de dados LDAP atualizado.

O arquivo **aliases** também ainda é a melhor maneira para definir listas de discussão (com a diretiva `:include:`). Os programas para lista de discussão normalmente direcionam a mensagem para um script de envoltório e a reenviam. Uma consulta LDAP pode retornar um endereço local que manipula lista de discussão (por meio do arquivo **aliases**), mas ela não é capaz de chamar um programa diretamente.

Na versão 8.12, o LDAP também pode ser usado para armazenar o conteúdo de outros dados de **sendmail** (por exemplo, tabelas e classes). Veja a seção 19.8 para obter maiores informações sobre como configurar **sendmail** para usar o LDAP e a seção 19.9 para um exemplo ligeiramente artificial que usa o LDAP para implementar nomes alternativos e hospedagem de domínios virtuais.

19.5 SENDMAIL: O DIRETOR DE ESPETÁCULOS DO CIRCO DE E-MAIL

Todas as três plataformas de referência Linux adotadas neste livro vêm com um agente para transporte de e-mails. O Red Hat e o SuSE fornecem o **sendmail** e o Debian usa Exim. Descrevemos rapidamente o Exim a partir da seção 19. 14. Muitas das construções e das opções de configuração do Exim possuem correspondentes no mundo **sendmail**. Já que este capítulo é muito longo, veremos somente **sendmail** de forma detalhada. Descrevemos o Exim em termos dos recursos equivalentes no **sendmail**.

sendmail pode adaptar-se aos caprichos daqueles que criam os padrões graças em parte à flexibilidade de seu arquivo de configuração, que permite que **sendmail** atenda às necessidades de uma comunidade de usuários diversa. O restante deste capítulo é dedicado basicamente ao entendimento e à construção de seu arquivo de configuração, o infame **sendmail.cf**.

sendmail é um agente de transporte, um programa que faz a interface entre os agentes de usuário e os agentes de entrega. Ele fala o protocolo SMTP e entrega mensagens para correspondentes em máquinas remotas via Internet. A lista de tarefas de **sendmail** abrange:

- Controlar mensagens à medida que elas deixam o teclado do usuário
- Compreender os endereços dos receptores
- Escolher um agente de transporte ou de entrega apropriado
- Reescrever endereços para um formato que possa ser compreendido pelo agente de entrega
- Reformatar cabeçalhos conforme a necessidade
- Passar a mensagem transformada para o agente de entrega

sendmail também gera mensagens de erro e retorna mensagens para o remetente caso não possam ser entregues.

www.snow666.blogspot.com
Snow666

A história de sendmail

sendmail V5 foi criado por Eric Allman em 1983. Uma ramificação foi otimizada por Lennart Lövstrand da Universidade de Linköping, na Suécia em 1987, e batizada de IDA **sendmail**[19]. Ela era mantida por Neil Rickert e Paul Pomes. Uma outra ramificação, King James Sendmail, foi desenvolvida por Paul Vixie na DECWRL durante o período de 1989 a 1993. Ela era baseada no IDA **sendmail**, mas tinha uma ênfase bem maior em vazão e desempenho para sites comerciais. O IDA e o KJS foram pioneiros em diversos recursos que agora são incluídos no **sendmail** V8, a principal reformulação de Eric datada de 1993.

Na época em que este livro era escrito, a maioria dos fornecedores Linux oferecia versões do **sendmail** derivadas do V8. Entretanto, elas estão tipicamente uma versão ou duas atrás da versão principal da Sendmail, Inc. Os fornecedores normalmente personalizam uma determinada versão de **sendmail** e, então, ficam relutantes em atualizar seus sistemas-base para incluir revisões atuais. Veja a Tabela 19.5 para obter informações sobre as versões de **sendmail** oferecidas com os diversos sistemas.

Baseamos nossa discussão de **sendmail** na V8.12 e ignoramos totalmente tanto as versões V5 como a IDA, que são obsoletas. A V8 usa o processador de macros **m4** para permitir uma fácil configuração dos casos-padrão. Essa "configuração leve" é tudo do que a maioria dos sites precisa.

Infelizmente, se a sua configuração apresentar algum problema, talvez você tenha de basear a sua depuração numa compreensão do arquivo de configuração bruto, que temos ouvido ser descrito como inacessível, atemorizante, enjoado, criptográfico, oneroso, infame, enfadonho, sádico, confuso, tedioso, ridículo, obscuro e enrolado. Falamos um pouco do arquivo de configuração em versões mais antigas deste livro, mas já que sua relevância para administradores modernos se esvaiu, agora indicamos o livro da O'Reilly sobre **sendmail** de Bryan Costales e Eric Allman (intitulado *sendmail*) ou o *Sendmail Installation and Operations Guide* que é incluído na distribuição **sendmail**.

Versões fornecidas pelos fabricantes de sendmail

A Tabela 19.5 lista a versão do **sendmail** que acompanha os sistemas Red Hat e SuSE ou disponíveis via **apt-get** em sistemas Debian. Ela também mostra onde nossos fornecedores "camaradas" esconderam o binário de **sendmail** e seus arquivos de configuração, **sendmail.cf** e **submit.cf**. Listamos sendmail.org como um forneceder para que você possa comparar os seus padrões caso substitua a versão de seu fornecedor. Em vez de ignorarmos o **sendmail** no Debian, que usa o Exim como agente de transporte, instalamos **sendmail** por meio de **apt-get** e incluímos estes detalhes.

Tabela 19.5 Versões de sendmail de diversos fornecedores (por volta de 2001).

Sistema	Código	Configuração	Diretório binário	Diretório de configuração
sendmail.org	8.12.2	Nível 10	–	**/etc/mail**
Red Hat 7.2	8.11.6	Nível 9	**/usr/sbin**	**/etc**
SuSE 7.3	8.11.2	Nível 9	**/usr/sbin**	**/etc**
Debian 3.0	8.9.3	Nível 8	**/usr/sbin**	/etc/mail

Novas versões de **sendmail** são lançadas algumas vezes para corrigir problemas relacionados com segurança; sugerimos que você verifique as notas de liberação ("release notes") em www.sendmail.org e atualize o seu sistema caso tenha perdido qualquer correção referente a questões de segurança.Você precisará de um compilador C e do pré-processador de macros **m4** (ambos normalmente inclusos nas distribuições Linux). Os fornecedores (distribuições) também fornecem correções de segurança para os pacotes que distribuem; assim, os sites das distribuições devem também ser verificados para correções de segurança, já que pode ser mais apropriado utilizar a correção específica da distribuição utilizada.

Algumas vezes é difícil determinar exatamente a verdadeira versão-base de **sendmail**, mas se o seu fornecedor não remexeu muito, você será capaz de executar

```
/usr/sbin/sendmail -d0.1 -bt < /dev/null
```

para fazer com que **sendmail** revele o número de sua versão, as opções que nela foram compiladas e quem ela acredita ser após ler o arquivo de configuração. O flag **-d** configura um nível de depuração (veja a página 462 para obter maiores informações sobre níveis de depuração em **sendmail**), o flag **-bt** coloca **sendmail** no modo de teste de en-

19. Lennart era um estudante do departamento de ciências de computação que na Suécia chama-se Institutionenför Datavetenskap; daí a origem do nome IDA.

Snow666

dereços e o redirecionamento a partir de **/dev/null** não dá a ele nenhum endereço para teste. Eis um exemplo de saída gerada (ligeiramente truncada) de um sistema Red Hat.

```
Version 8.11.6
Compiled with: LDAPMAP MAP_REGEX LOG MATCHGECOS MIME7TO8
   MIME8TO7 NAMED_BIND NETINET NETINET6 NETUNIX NEWDB NIS
   QUEUE SASL SCANF SMTP TCPWRAPPERS USERDB
============ SYSTEM IDENTITY (after readcf) ============
    (short domain name) $w = coyote
(canonical domain name) $j = coyote.toadranch.com
        (subdomain name) $m = toadranch.com
             (node name) $k = coyote.toadranch.com
=======================================================
```

sendmail sempre deve usar registros DNS MX (mail exchanger) e fará desta maneira se compilado com a opção `NAMED_BIND` (como no exemplo anterior). As variáveis de uma letra como `$w` são do arquivo de configuração bruto ou determinadas em tempo de execução.

Instalação de sendmail em sendmail.org

O ambiente de instalação de **sendmail** versão 8.12 mudou um pouco. **sendmail** não executa mais setuid para root, mas sim setgid para o grupo de sendmail, smmsp. Antes de instalar **sendmail**, você tem de criar o usuário smmsp e o grupo smmsp (o nome misterioso significa **s**end**m**ail **m**ail **s**ubmission **p**rogram). Usamos UID e GID 25 para concordar com o bem conhecido número de porta do protocolo de correio SMTP. O usuário smmsp deve ter smmsp como seu grupo de login padrão, que é configurado tipicamente no arquivo **/etc/passwd**. A adição de um usuário e grupo sendmail dedicados permite que **sendmail** seja executado com privilégios reduzidos e aumenta a segurança.

A segunda maior alteração sob o ponto de vista do administrador é que **sendmail** agora usa dois arquivos de configuração: **submit.cf** e **sendmail.cf**. O arquivo **submit.cf** controla a manipulação de e-mail que se origina na máquina local (e está sendo inicialmente submetido ao sistemas de e-mail) e **sendmail.cf** controla e-mails que chegam e e-mails colocados em fila durante o processo de submissão. **submit.cf** é fornecido com a distribuição e é o mesmo para todos os sites; normalmente não há nenhuma necessidade de personalizá-lo.

Esta seção descreve brevemente o processo de instalação; consulte as notas de instalação na distribuição **sendmail** para detalhes e para questões relacionadas com arquiteturas ou sistemas operacionais em particular. A próxima seção descreve a instalação de **sendmail** num sistema Debian via **apt-get**. Se estiver substituindo a versão original de **sendmail** de seu fornecedor, algumas das tarefas de configuração (como instalação dos arquivos de ajuda) talvez já tenham sido feitas para você.

Os envolvidos:

- O binário de **sendmail**, normalmente instalado em **/usr/sbin**.
  `-r-xr-sr-x  root  smmsp  /usr/sbin/sendmail`
- Os arquivos de configuração **/etc/mail/sendmail.cf** e (nas versões 8.12 e posteriores) **/etc/mail/submit.cf**, instalados pelo administrador do sistema:
  `-rw-r--r--  root  bin  /etc/mail/sendmail.cf`
  `-rw-r--r--  root  bin  /etc/mail/submit.cf`
- Os diretórios de filas de e-mail, **/var/spool/mqueue** e (nas versões 8.12 e superiores) **/var/spool/clientmqueue**, criados pelo administrador do sistema ou pelo processo de instalação:
  `drwxrwx---  smmsp  smmsp  /var/spool/clientmqueue`
  `drwx-----  root  wheel  /var/spool/mqueue`
- Vários links para o binário de **sendmail** (**newaliases**, **mailq**, **hoststat** etc.)[20]
- Agentes de entrega locais de **sendmail** mais seguros, **smrsh** e **mail.local** (normalmente instalados em **/usr/libexec**)

Você pode fazer o download da versão mais recente de **sendmail** no endereço www.sendmail.org. Para compilar e instalar o pacote, siga as instruções contidas no arquivo **INSTALL** no diretório-base. Comece adicionando o usuário e grupo smmsp; não dê a este usuário um shell verdadeiro. Eis uma entrada **/etc/passwd** típica:

20. Tome cuidado aqui. Alguns fornecedores usam links "hard", cuja conseqüência é que ao fazer uma atualização, talvez você acabe terminando com uma versão de **sendmail** não correspondente à versão de **newaliases**, criando dores de cabeça sutis e difíceis de serem encontradas.

Snow666


```
smmsp:x:25:25:Sendmail Mail Submission Prog:/nao-existente:/bin/false
```

E eis uma entrada **/etc/group** típica:

```
smmsp:*:25:smmsp
```

Para compilar o software, mude para o diretório de distribuição de **sendmail**, execute o script **Build** e a seguir execute **Build install**. O arquivo **devtools/OS/Linux** contém os procedimentos usados para construir **sendmail** em sistemas Linux. As distribuições Linux não padronizaram onde as coisas devem se encontrar, de forma que **devtools/OS/Linux** é um local provável, mas que talvez não seja exatamente correto para o caso da sua distribuição.

Entretanto, antes de começar a compilar, você deve decidir-se sobre um formato de banco de dados e uma estratégia para interfacear **sendmail** com bancos de dados administrativos como o NIS. Para bancos de dados instalados em disco, recomendamos o pacote Berkeley DB especificado no **Makefile** como `NEWDB` (o padrão). O Linux vem com uma versão antiga do Berkeley DB (safra 1994). Embora **sendmail** opere com essa antiga versão, recomendamos que você obtenha uma versão atual em sleepycat.com. Os termos para licenciamento são bem generosos: desde que você não distribua o software DB, não é necessária nenhuma licença e não há nenhuma taxa a ser paga.[21]

Para personalizar o Makefile, não o edite; crie o seu próprio arquivo **site.config.m4** e coloque-o no diretório **devtools/Site** para ajustá-lo ao seu sistema operacional e ambiente local. Por exemplo, se você pretende usar o LDAP e a nova biblioteca para filtragem de spam, crie nesse diretório um arquivo **site.config.m4** contendo as linhas

```
APPENDDEF(`confMAPDEF', `-DLDAPMAP')
APPENDDEF(`confLIBS', `-lldap -llber')
APPENDDEF(`conf_sendmail_ENVDEF', `-DMILTER')
```

Um `define` substitui a definição atual de um atributo; a macro `APPENDDEF` anexa à definição atual.

Compilar **sendmail** com

```
% sh ./Build
```

irá incluir automaticamente as entradas específicas ao site. Para instalar **sendmail** no local apropriado, execute

```
% sudo sh ./Build install
```

sendmail normalmente *não* deve ser configurado para ser controlado por **inetd/xinetd**, de modo que ele precisa ser iniciado explicitamente nos arquivos **rc** no momento da inicialização. Uma seqüência típica ficaria parecida com o seguinte:

```
if [-f /usr/sbin/sendmail -a -f /etc/mail/sendmail.cf];
then
   (cd /var/spool/clientmqueue; rm -f [tTx]f*)
   (cd /var/spool/mqueue; rm -f [tTx]f*)
   /usr/sbin/sendmail -bd -q30m          ### executor de fila para a fila normal
   /usr/sbin/sendmail -Ac -q30m &        ### executor de fila para a fila de clientes (8.12)
   echo -n ' sendmail' > /dev/console
fi
```

Essas linhas verificam a existência do binário de **sendmail** e seu arquivo de configuração e depois iniciam o programa no modo daemon.

Vários agentes de usuário executam **sendmail** explicitamente (algumas vezes com os flags **-bm** ou **-bs**) ao submeterem a mensagem de um usuário ao sistema de e-mails em vez de falarem o protocolo SMTP diretamente. Nesse caso, **sendmail** usa o arquivo de configuração **submit.cf** e coloca as mensagens no diretório de fila **/var/spool/clientqueue**. Chamar **sendmail** com os flags **-Am** ou **-Ac** força as mensagens para, respectivamente, **mqueue** ou **clientqueue**.

O Red Hat possui um script de inicialização de **sendmail** (**/etc/init.d/sendmail**) que não limpa o lixo dos diretórios **mqueue** (como demonstrado no exemplo anterior). Entretanto, ele efetivamente reconstrói os mapas de bancos de dados. Com o **sendmail** 8.12 e posterior, o LDAP pode ser usado para mapas de bancos de dados, e reconstruir os mapas após alterações não é necessário. Parâmetros definidos em **/etc/sysconfig/sendmail** determinam se **sendmail** deve ser iniciado no modo daemon e com que freqüência deve executar a fila, assim, é lá que você deve configurar o comportamento de inicialização de **sendmail** para o seu site. O Red Hat, conforme distribuído, inicia **sendmail** no modo daemon e executa a fila a cada 30 minutos.

---

21. A licença na verdade especifica que pode haver redistribuição desde que o código-fonte também seja disponível em condições razoáveis (N.R.T.).

Snow666

O script de inicialização de SuSE (**/etc/init.d/sendmail**) simplesmente verifica a existência do binário e do arquivo de configuração e depois inicia **sendmail** com os argumentos definidos na variável de ambiente SENDMAIL_ARGS; estes são configurados no arquivo **/etc/rc.config.d/sendmail.rc.config**. O SuSE também tem como padrão o modo daemon e executa a fila a cada 30 minutos.

O fragmento **sh** apresentado (ou um como este) deve ser adicionado a **/etc/init.d/sendmail** caso sua distribuição Linux não inclua um script de inicialização **sendmail**. Um script mais exagerado no guia de instalação tenta limpar as execuções de filas previamente interrompidas. Misture e componha à vontade.

Historicamente, os arquivos de suporte de **sendmail** ficaram perambulando em torno do sistema de arquivos em destinos glamourosos como **/usr/lib**, **/etc**, **/usr/ucblib** e **/usr/share**. Com a versão 8.10 de **sendmail**, todos os arquivos (exceto os diretórios de filas) supostamente devem estar sob o mesmo diretório **/etc/mail**.[22] Esperemos que os fornecedores aceitem a dica e os deixem em um único local consistente.

Instalando o sendmail em sistemas Debian

Você pode usar o programa **apt-get** do Debian para instalar **sendmail**. Ele instala **sendmail** (versão 8.9.3[23]) e **m4** e também desinstala Exim. Depois de **apt-get** ter feito o download e instalado o pacote **sendmail** do Debian, ele se oferecerá para configurar o pacote. Responder yes chamará um script que lhe fará algumas perguntas (cerca de 20) sobre a configuração desejada para **sendmail**. Cada pergunta mostra a resposta-padrão entre colchetes e, exceto algumas exceções, lhe dará um arquivo de configuração razoável. A única pergunta que você realmente deve responder de forma diferente do valor-padrão é "mail name"; a resposta-padrão é o nome de host não completamente qualificado (por exemplo, lappie), mas ele precisa ser o nome completamente qualificado (por exemplo, lappie.toadranch.com).

Responder às perguntas feitas pelo script fará mais sentido caso você já esteja familiarizado com as opções e os recursos de **sendmail**, que são discutidos de forma mais detalhada ainda neste capítulo. O script de configuração não leva em consideração o conteúdo do arquivo de domínio genérico que ele inclui como padrão. Conseqüentemente, você pode não aceitar um recurso, mas acabar com ele ativado de qualquer maneira (por exemplo, o recurso `redirect`).

A comutação de serviços é vista em detalhes no Capítulo 18.

O arquivo de comutação

Os sistemas Linux possuem um arquivo de configuração de "comutação de serviços", **/etc/nsswitch.conf**, que enumera os métodos que podem ser usados para satisfazer diversas consultas-padrão como pesquisas de host e de usuário. Se mais de um método de resolução for listado para um determinado tipo de consulta, o comutador de serviços também determinará a ordem na qual os métodos são consultados.

O emprego da comutação de serviços normalmente é transparente para o software. Entretanto, **sendmail** gosta de exercer um controle mais refinado sobre as pesquisas, de modo que ele atualmente ignora o arquivo de comutação do sistema e usa o seu próprio arquivo interno (**/etc/mail/service.switch**) em seu lugar.

Dois campos no arquivo de comutação impactam o sistema de e-mail: `aliases` e `hosts`. Os valores possíveis para o serviço de hosts são `dns`, `nis`, `nisplus` e `files`. Para nomes alternativos, os valores possíveis são `files`, `nis`, `nisplus` e `ldap`. Devem ser compilados em **sendmail** suporte para todos os mecanismos usados (exceto `files`) antes de o serviço poder ser usado.

O arquivo de comutação de serviços interno de **sendmail** contém o seguinte:

```
aliases     files nisplus nis        # if compiled with nis/nis+ (Se compilado com xis/nis+)
hosts       dns nisplus nis files
```

Modos de operação

Você pode executar **sendmail** em diversos modos, selecionados por meio do flag **-b**. **-b** significa "be" ou "become" (ou seja, "ser" e "tornar-se") e é sempre usado com um outro flag que determina o papel que **sendmail** irá desempenhar. A Tabela 19.6 lista os valores válidos.

22. Bem, isso não é totalmente verdadeiro apesar de todos os arquivos serem mantidos sob **/etc/mail**. O arquivo **sendmail.pid** e algumas vezes o arquivo de estatísticas ainda são mantidos em algum outro local.
23. Na versão 3.0r1 do Debian, o **sendmail** está na versão 8.12.3, com diversas atualizações de segurança disponíveis em http://security.debian.org (N.R.T.)

www.snow666.blogspot.com
Snow666

Tabela 19.6 Flags em linha de comandos para os principais modos de sendmail

Flag	Significado
-bd	Executar no modo daemon, escutar conexões na porta 25
-bD	Executar no modo daemon, mas em primeiro plano em vez de segundo plano[a]
-bh	Ver as informações de conexão recentes (o mesmo que **hoststat**)
-bH	Eliminar a cópia de disco de informações de conexões desatualizadas (o mesmo que **purgestat**)
-bi	Inicializar nomes alternativos hashed (o mesmo que **newaliases**)
-bm	Executar como entregador de e-mail; entregar e-mails da maneira usual (padrão)
-bp	Imprimir a fila de e-mails (o mesmo que **mailq**)
-bP	Imprimir o número de entradas nas filas via memória compartilhada (versões 8.12 e posteriores)
-bs	Entrar no modo servidor SMTP (na entrada-padrão, e não na porta 25)
-bt	Entrar no modo de teste de endereços
-bv	Verificar somente endereços de e-mail; não enviar e-mails

a. Este modo é usado para depuração, para você poder ver mensagens de erro e de depuração.

Se você espera que os e-mails provenham da Internet, execute **sendmail** no modo daemon (**-bd**). Nesse modo, **sendmail** escuta a porta de rede 25 e aguarda por trabalho.[24] Normalmente você também especificará o flag **-q** — ele configura o intervalo no qual **sendmail** processa a fila de e-mails. Por exemplo, **-q30m** executa a fila a cada trinta minutos e **-q1h** a executa a cada hora.

sendmail normalmente tenta entregar as mensagens imediatamente, salvando-as na fila apenas momentaneamente para garantir confiabilidade. Mas se o seu host for muito ativo ou a máquina de destino estiver inacessível, **sendmail** colocará a mensagem em fila e tentará enviá-la novamente mais tarde. **sendmail** costumava usar um processo-filho cada vez que processava a fila, mas agora ele suporta executores de filas persistentes que normalmente são inicializados no momento da inicialização. A RFC1123 recomenda pelo menos 30 minutos entre as execuções. **sendmail** realiza bloqueio, de modo que várias execuções de filas simultâneas são seguras.

sendmail 8.12 acrescentou um novo recurso para ajudar no caso de grandes listas de discussão e grandes filas: grupos de filas com divisão de envelopes. Ele é visto de forma mais detalhada a partir da página 458.

sendmail lê o seu arquivo de configuração, **sendmail.cf**, somente ao inicializar. Portanto, você tem de abortar e reiniciar **sendmail** ou, então, enviar a ele um sinal HUP ao alterar o arquivo de configuração. **sendmail** cria um arquivo **sendmail.pid** contendo o ID de processo e o comando que o iniciou. Você deve iniciar **sendmail** com um caminho absoluto, pois ele se auto-executa novamente ao receber o sinal HUP. O arquivo **sendmail.pid** permite que o processo receba o sinal HUP via:

```
# kill - UP  head -1 sendmail.pid
```

A localização do arquivo PID costumava ser um parâmetro definido em tempo de compilação, mas agora ele pode ser configurado no arquivo de configuração **.mc** pela opção `confPID_FILE`.

```
define(confPID_FILE, `/var/run/sendmail.pid')
```

O valor-padrão depende do sistema operacional, mas usualmente é **/var/run/sendmail.pid** ou **/etc/mail/sendmail.pid**. O Red Hat e o Debian usam **/var/run/sendmail.pid** e o SuSE o mantém no diretório **/var/run/sendmail**.

A fila de e-mails

As mensagens de e-mail são ordenadas no diretório de filas quando a máquina está muito ocupada para poder entregá-las imediatamente ou quando a máquina de destino se encontra indisponível. **sendmail** serve como um agente de submissão de e-mails escutando a porta 587, bem como cumprindo seu papel tradicional, ou seja, como um daemon escutando na porta 25 e usando a fila **/var/spool/mqueue**. Alguns agentes de usuário (**mh** e **/bin/mail**, por exemplo) usam a porta 587, a porta de submissão de e-mails, mas outros (Eudora, Outlook etc.) falam SMTP diretamente com o **sendmail** executando na porta 25. A partir da versão 8.12, programas de submissão de e-mails introduzem novas mensagens no sistema de e-mails usando o diretório de filas **/var/spool/clientmqueue** e o arquivo de configuração **submit.cf**. Todas as mensagens vão para a fila brevemente à medida que chegam.

24. As portas que **sendmail** escuta são determinadas por `DAEMON_OPTIONS`; a porta 25 é o padrão.

www.snow666.blogspot.com
Snow666

sendmail permite que você tenha mais de uma fila de e-mails e identifique subconjuntos das filas como pertencentes a um grupo de filas. Por exemplo, se o diretório **mqueue** contivesse os subdiretórios **q1**, **q2** e **q3** e você especificasse o diretório de filas como **/var/spool/mqueue/q***, todas as três filas seriam usadas. A capacidade de **sendmail** em manipular várias filas aumenta o desempenho sob alta carga.[25] Se um site estiver executando uma lista de discussão grande, **sendmail** dividirá a lista receptora de envelopes em diversas listas menores e as atribuirá a diferentes grupos de filas. Esse truque pode melhorar muito o desempenho, pois as listas de receptores menores podem ser processadas em paralelo.

Os grupos de filas são novos na versão 8.12 e dão um controle mais refinado sobre tipos de mensagens individuais. Qualquer um dos parâmetros associados a filas também pode ser configurado num determinado grupo de filas, incluindo a prioridade de execução (com a chamada de sistema **nice**). Os e-mails são submetidos a um grupo de filas de acordo com o endereço do primeiro receptor da mensagem. O grupo de filas padrão chama-se `mqueue` e é definido automaticamente e disponível para uso sem configuração adicional. Grupos de filas são abordadas em detalhe na seção 19.12.

Quando uma mensagem é colocada em fila, ela é salva em pedaços em vários arquivos diferentes. Cada nome de arquivo possui um prefixo de duas letras que identifica o trecho, seguido de um ID aleatório formado do ID de processo de **sendmail**. Esse ID não é fixo, pois **sendmail** está criando processos constantemente e cada cópia obtém um novo ID de processo. A Tabela 19.7 ilustra os seis pedaços possíveis.

Se os subdiretórios **qf**, **df** ou **xf** existirem num diretório de filas, então estes pedaços da mensagem serão colocados no subdiretório apropriado. O arquivo **qf** contém não somente o cabeçalho da mensagem, como também os endereços de envelope, a data na qual a mensagem deveria ser retornada quando não podendo ser entregue, a prioridade da mensagem na fila e a razão da mensagem estar na fila. Cada linha começa com um código de uma só letra que identifica o restante da linha.

Tabela 19.7 Prefixos para arquivos na fila de e-mails.

Prefixo	Conteúdo dos arquivos
qf	O cabeçalho da mensagem e o arquivo de controle
df	O corpo da mensagem
tf	Uma versão temporária do arquivo **qf** enquanto o arquivo **qf** estiver sendo atualizado
Tf	Significa que ocorreram 32 ou mais tentativas falhas de bloqueio
Qf	Significa que a mensagem ricocheteou e não pôde ser retornada
xf	Arquivo de transcrição temporário das mensagens de erros dos entregadores de e-mail

Cada mensagem que é colocada na fila deve ter um arquivo **qf** e **df**. Todos os demais prefixos são usados por **sendmail** durante uma tentativa de entrega. Quando uma máquina trava e reinicializa, a seqüência de inicialização para **sendmail** deve eliminar os arquivos **tf**, **xf** e **Tf** de cada diretório de filas. O administrador de sistemas responsável pelos e-mails deve verificar ocasionalmente a existência de arquivos **Qf** no caso de a configuração local estar provocando os ricochetes.

A fila de e-mails fornece diversas oportunidades para as coisas darem errado. Por exemplo, o sistema de arquivos pode ficar lotado (evite colocar **/var/spool/mqueue** e **/var/spool/news** na mesma partição), a fila pode se tornar obstruída, e mensagens de e-mails órfãs podem ficar emperradas na fila.

sendmail possui uma opção de configuração (`confMIN_FREE_BLOCKS`) para auxiliar a gerenciar o espaço em disco. Quando o sistema de arquivos contendo a fila de e-mails ficar muito cheio, os e-mails serão rejeitados com uma mensagem de erro "try again later" (ou seja, tente mais tarde) até que haja mais espaço disponível. Essa opção deixa um pouco de espaço sobrando de modo que os e-mails começam a ser rejeitados antes de o sistema de arquivos estar totalmente cheio.

Consulte a seção 16.11 para obter maiores informações sobre os registros MX do DNS.

Se um centralizador de e-mail importante cair, seus sites backup MX poderão ficar sobrecarregados com milhares de mensagens. **sendmail** poderá criar muitas cópias de si mesmo e levar uma máquina a cair. Várias opções ajudam no desempenho em máquinas muito ativas; reunimos estas na seção 19. 12, sobre desempenho.. Para manipular uma fila temporariamente obstruída antes da versão 8.12, você tinha que mover a fila obstruí-

25. Diretório é um mecanismo de armazenamento eficiente se ele não contiver muitos arquivos. Se você tiver um servidor de e-mails com muitas listas de discussão que se tornam defasadas, o diretório de filas pode facilmente se tornar tão grande que não possa ser manipulado com eficiência.

www.snow666.blogspot.com
Snow666

da, continuar processando novos e-mails como de praxe e executar uma cópia distinta de **sendmail** na fila obstruída depois que as coisas se acalmassem. Por exemplo, o procedimento para manipular um único diretório de filas seria:

```
# kill  head -1 sendmail.pid
# mv mqueue cloggedqueue          /* Para um outro FS, se necessário */
# mkdir mqueue                    /* Configurar também proprietários/permissões */
# chown root mqueue
# chmod 700 mqueue
# /usr/sbin/sendmail -bd -q1h
```

Quando as coisas se acalmarem, execute **sendmail** com os seguintes flags:

```
# /usr/sbin/sendmail -oQ/var/spool/cloggedqueue -q
```

Esses flags apontam **sendmail** para o diretório da fila obstruída e especificam que **sendmail** deve processá-lo imediatamente. Repita esse comando até que a fila se esvazie. A partir da versão 8.12, **sendmail** usa hard links em modos que irão quebrar caso você desloque uma fila. Uma melhor maneira de lidar com filas obstruídas é usar uma máquina de emergência e registros MX; veja a seção sobre desempenho para maiores detalhes.

O ponto no qual a fila se torna obstruída depende do site e do hardware no qual **sendmail** está sendo executado. O seu sistema e o centralizador de e-mail para aol.com, que processa milhões de mensagens por dia, terão visões diferentes do que é uma fila. Veja a seção 19.13 para obter informações sobre como medir níveis de tráfego.

19.6 CONFIGURAÇÃO DE SENDMAIL

Antes da versão 8.12, as ações de sendmail eram controladas por um único arquivo de configuração, **/etc/mail/sendmail.cf** (ele era encontrado anteriormente em **/etc** ou **/usr/lib**). Nós o chamamos de arquivo config para fins de abreviação. A versão 8.12 introduziu uma segunda instância do arquivo de configuração chamado **submit.cf** (também no diretório **/etc/mail**). Os flags com os quais **sendmail** é iniciado determinam qual arquivo de configuração ele usará: **-bm**, **-bs** e **-bt** usam **submit.cf** caso ele exista, e todos os demais modos usam **sendmail.cf**. Obviamente, há flags de linha de comando e opções de arquivos config que mudam os nomes dos arquivos de configuração, porém é melhor deixá-los de lado. O arquivo de configuração determina o seguinte para **sendmail**:

- Escolha de agentes de entrega
- Regras para reescrita de endereços
- Formatos de cabeçalhos de e-mails
- Opções
- Precauções quanto à segurança
- Resistência a spams

O formato do arquivo de configuração bruto foi concebido para ser fácil de analisar sintaticamente. Esse foco o deixou um tanto carente em termos de recursos amigáveis. A manutenção do arquivo de configuração é a tarefa administrativa mais importante relacionada com os e-mails e aterroriza até mesmo os administradores de sistemas mais experientes.

Toda versão de **sendmail** usa um arquivo de configuração, porém as versões modernas facilitam o processo de configuração pelo emprego de macros **m4**, que ocultam grande parte da complexidade subjacente. Pode-se dizer que o arquivo de configuração bruto se encontra em nível de linguagem assembly, ao passo que a configuração via **m4** está mais para o nível do Perl.[26]

Quando as macros **m4** foram introduzidas pela primeira vez, esperava-se que elas fossem capazes de manipular de 80% a 90% dos casos. De fato, a taxa de cobertura acabou sendo até maior, provavelmente perto dos 98%. Neste livro, abordamos somente o "arquivo de configuração leve" baseado em macros **m4**. Você precisará se aprofundar no arquivo de configuração de baixo nível somente se estiver depurando um problema espinhoso, ampliando o seu site de e-mail de modo bizarro ou executando um centralizador de e-mail de altíssimo volume.

Três documentos importantíssimos são o livro da O'Reilly, *sendmail*, de Bryan Costales e Eric Allman, o artigo *Sendmail Installation and Operations Guide*, de Eric Allman (incluso no diretório **doc/op** da distribuição), e o arquivo **README** (no diretório **cf**). Normalmente nos referimos a *sendmail* como uma fonte para maiores informa-

26. A linguagem de configuração de **sendmail** é "completa em termos da máquina de Turing", o que significa que ela pode ser usada para escrever qualquer programa de computação possível. Os leitores que se depararam com o arquivo de configuração bruto entenderão quão assustador é esse conceito ...

www.snow666.blogspot.com
Snow666

ções e nos referimos a ele como "o livro do **sendmail**". Da mesma forma, nos referimos ao artigo sobre instalação como "o guia de instalação" e ao arquivo README como **cf/README**.

Utilizando o pré-processador m4

Primeiramente descrevemos alguns recursos **m4**, mostramos como construir um arquivo de configuração por meio de um arquivo mestre **m4** e, finalmente, descrevemos algumas importantes macros **m4** pré-empacotadas advindas com a distribuição de **sendmail**. Nem o Red Hat nem o SuSE incluem a página de manual referente a **m4** em suas distribuições, embora o comando esteja lá. O Debian instala tanto **m4** quanto sua página de manual ao instalar **sendmail** com **apt-get**.

Concluímos esta seção com exemplos de configuração para três instalações distintas:

- Um computador Linux doméstico de um estudante de computação
- Uma empresa de porte médio que sabe como configurar **sendmail** de maneira apropriada
- Um site que tem bastante hospedagem Web

Também mostramos as configurações-padrão fornecidas com as distribuições Linux Red Hat, SuSE e Debian.

m4 destinava-se originalmente como um front-end para linguagens de programação que deixaria o usuário gravar programas mais legíveis (ou quem sabe mais criptográficos). **m4** é suficientemente poderoso para ser útil em diversas situações de transformação de entradas e ele funciona perfeitamente para arquivos de configuração de **sendmail**.

As macros **m4** assumem a seguinte forma

```
nome(arg1, arg2, ..., argn)
```

Não deve haver nenhum espaço entre o nome e o parênteses de abertura. Aspas únicas de abertura e de fechamento são usadas para envolver strings e defini-las como argumentos. As convenções de colocação de aspas das macros **m4** são diferentes daquelas de outras linguagens que você possa ter usado, já que as aspas de abertura e de fechamento são caracteres diferentes.[27] Aspas podem ser aninhadas, também. Com as ferramentas de construção dos compiladores atuais, alguém pode ficar imaginando como o **m4** sobreviveu com uma sintaxe tão rígida e exótica.

m4 possui algumas macros embutidas, e os usuários também podem definir as suas próprias. A Tabela 19.8 apresenta as macros embutidas mais comuns usadas na configuração de **sendmail**.

Alguns sites adicionam uma macro `dnl` no final de cada linha para manter mais organizado o arquivo **.cf** convertido; sem `dnl`, **m4** adiciona linhas em branco extras ao arquivo de configuração. Essas linhas em branco não afetam o comportamento de **sendmail**, mas tornam o arquivo de configuração mais difícil de ser lido. Omitimos `dnl` de nossos exemplos. Outros sites usam `dnl` no início de suas linhas destinadas a comentários.

Tabela 19.8 macros m4 comumente usadas com sendmail.

Macro	Função
`define`	Define uma macro chamada arg1 com o valor arg2
`undefine`	Descarta uma definição de macro anterior chamada arg1
`include`	Inclui (interpola) o arquivo chamado arg1
`dnl`	Descarta caracteres até e inclui a próxima nova linha
`divert`	Gerencia fluxos de saída

m4 realmente não honra os comentários em arquivos. Um comentário como:

```
# E depois define a _
```

não faria o que você esperaria, pois `define` é uma palavra-chave **m4** e seria expandida. Em vez dela, use a palavra-chave `dnl` de m4 (que significa "delete to newline", "delete até a nova linha"). Por exemplo,

```
dnl # E depois define a ...
```

funcionaria. Você teria de colocar um espaço após `dnl` ou então um sinal de pontuação para ele ser reconhecido como um comando **m4**.

27. Você pode modificar os caracteres para envolvimento com a macro `changequote`, mas agir desta forma romperia totalmente o emprego do **m4** em **sendmail** devido a várias macros fazerem suposições sobre os caracteres de envolvimento.

Snow666

As peças de configuração de sendmail

A distribuição **sendmail** inclui um subdiretório **cf** contendo todas as peças necessárias para a configuração de **m4**: um arquivo **README** e vários subdiretórios, listados na Tabela 19.9.

Tabela 19.9 Subdiretórios de configuração.

Diretório	Conteúdo
cf	Modelos de arquivos **.mc** (master configuration, configuração mestra)
domain	Modelos de arquivos **m4** para vários domínios na Berkeley
feature	Fragmentos que implementam diversos recursos
hack	Recursos especiais de valor ou implementação dúbia
m4	O arquivo de configuração básico e outros arquivos importantes
ostype	Localizações e peculiaridades de arquivos dependentes do sistema operacional
mailer	Arquivos **m4** que descrevem mailers comuns (agentes de entrega)
sh	Scripts de shell usados por **m4**

O diretório **cf/cf** contém exemplos de arquivos **.mc**. De fato, ele contém tantos exemplos que os seus se perderiam no meio desse entulho. Recomendamos que você mantenha seus arquivos **.mc** separados do diretório **cf** distribuído. Crie um diretório nomeado para o seu site (**cf/***nome-do-site*) ou mude o diretório **cf** para **cf.examples** e crie um diretório **cf**. Se você fizer isso, copie o **Makefile** e o script **Build** para o seu novo diretório para que as instruções no arquivo **README** ainda funcionem. É melhor também copiar todos os arquivos de configuração **.mc** para um local centralizado em vez de deixá-los dentro da distribuição **sendmail**. O script **Build** usa nomes de caminhos relativos que terão de ser modificados caso você tente criar um arquivo **.cf** por meio de um arquivo **.mc** e não esteja na hierarquia de distribuição.

Criando um arquivo de configuração com base em um modelo de arquivo .mc

Antes de mergulharmos nas páginas e páginas de detalhes sobre as diversas macros de configuração, recursos e opções, colocaremos o carro antes dos bois e criaremos uma configuração "sem ornamentações" para ilustrar o processo. Nosso exemplo é para um nó tipo folha (de extremidade), foo.com; o arquivo de configuração mestre é chamado **foo.mc**.

Colocaremos **foo.mc** em nosso novo e brilhante diretório **cf**. O arquivo de configuração convertido (por **m4**) será **foo.cf** no mesmo diretório e, finalmente, vamos instalá-lo como **sendmail.cf** em **/etc** ou **/etc/mail**. **/etc/mail** é o **sendmail** padrão para a localização do arquivo de configuração, mas diversas distribuições usam **/etc**.

Algumas frases-padrão devem ser colocadas em cada arquivo **.mc** novo:

```
divert(-1)
#### arquivo .mc básico para foo.com
divert(0)
VERSIONID(`$Id$')
```

Caso queira colocar comentários no início de seu arquivo, a primeira linha normalmente é uma instrução `divert`, que joga fora quaisquer informações inúteis espúrias dos fluxos de saída de **m4** e alivia a necessidade de `dnl` em comentários do tipo #. Os comentários vêm após, seguido de uma outra instrução `divert`. Uma linha `VERSIONID` (aqui, no formato RCS) completa as frases-padrão. Ele é descrito detalhadamente na próxima seção.

Em muitos casos, especificar um `OSTYPE` (veja a página 415) para acionar caminhos ou parâmetro dependentes de sistema operacional e também um conjunto de `MAILERs` (veja a página 417) completará a configuração:

```
OSTYPE(`linux')
define(`confCOPY_ERRORS_TO', `postmaster')
MAILER(`local')
MAILER(`smtp')
```

Nesse caso, também configuramos uma opção (`confCOPY_ERRORS_TO`) que envia uma cópia de cabeçalhos de qualquer e-mail ricocheteado para o postmaster local. Essa notificação permite que o postmaster intervenha quando o problema se encontrar no site local.

Snow666

Para criar o verdadeiro arquivo de configuração, basta executar o comando **Build** que você copiou para o novo diretório **cf**:

```
# ./Build foo.cf
```

ou então

```
# make foo.cf
```

Finalmente, instale **foo.cf** no local correto — normalmente **/etc/mail/sendmail.cf**, porém tanto o Red Hat quanto o SuSE o colocam em **/etc/sendmail.cf**.

Um site maior pode criar um arquivo **m4** separado para armazenar padrões válidos para todo o site no diretório **cf/domain**; hosts individuais então incluem o conteúdo deste arquivo. Nem todo host precisa de um arquivo de configuração separado, mas cada grupo de hosts similares (mesma arquitetura e mesmo papel: servidor, cliente etc.) provavelmente precisará de uma configuração própria.

Alterando a configuração de sendmail

Freqüentemente achará que a sua configuração de sendmail está quase perfeita, mas você gostaria apenas de experimentar um novo recurso, adicionar uma nova regra de spam ou fazer uma alteração simples. Para tanto:

- Edite o arquivo **.mc** e introduza as alterações desejadas.
- Reconstrua o arquivo de configuração com o script **Build** no diretório de configuração.
- Instale o arquivo **cf** resultante como **sendmail.cf** no diretório correto.
- Envie um sinal HUP para **sendmail** para fazer com que ele releia seu arquivo de configuração.[28]

Mesmo com as facilidades do novo sistema de configuração de **sendmail**, você ainda tem de tomar uma série de decisões referentes ao seu site. À medida que for lendo sobre os recursos descritos a seguir, imagine como eles se encaixariam na organização de seu site. Um site pequeno provavelmente terá somente um nó centralizador e nós de extremidade, portanto, precisará somente de duas versões do arquivo de configuração. Um site maior poderia precisar de centralizadores distintos para e-mails de entrada e de saída e, quem sabe, de um servidor POP/IMAP separado.

Seja lá qual for a complexidade de seu site e seja lá qual for o seu aspecto (exposto, por trás de um firewall ou numa rede virtual privada, por exemplo), é provável que o diretório **cf** contenha alguns trechos de configuração já prontos e adequados apenas aguardando que sejam personalizados e colocados em funcionamento.

19.7 Primitivas de Configuração Básicas para Sendmail

Os comandos de configuração para **sendmail** distinguem o emprego de letras maiúsculas/minúsculas. Por convenção, os nomes das macros predefinidas são todos em letras maiúsculas (por exemplo, `OSTYPE`), comandos **m4** são em letras minúsculas (por exemplo, `define`) e nomes de variáveis configuráveis começam com um `conf` em letras minúsculas e terminam com um nome de variável com todas as letras maiúsculas (por exemplo, `confCOPY_ERRORS_TO`). As macros normalmente se referem a um arquivo **m4** chamado **../*nomedamacro/arg1*.m4**. Por exemplo, a macro `OSTYPE(`linux')` faz com que **../ostype/linux.m4** seja incluído.

Nesta seção veremos os comandos de configuração básicos e deixaremos os recursos mais pomposos para mais tarde.

A macro VERSIONID

Você deve manter seus arquivos de configuração com CVS ou RCS, não somente para poder retornar para uma configuração anterior caso necessário, como também para poder identificar as versões dos arquivos **m4** que formarão o arquivo de configuração. Use a macro `VERSIONID` para embutir automaticamente informações relativas à versão. A sintaxe é

```
VERSIONID(`$Id$')
```

As informações de versão em si serão preenchidas pelo RCS à medida que você registra o arquivo. Ela aparecerá no arquivo **sendmail.cf** final na forma de um comentário. Essas informações também podem ser úteis caso você

28. Use o comando **kill** para fazer isso. O arquivo **sendmail.pid** facilita o processo de localização do ID de processo de **sendmail**; infelizmente, sua localização não é consistente entre as distribuições (tente **/var/run/sendmail.pid**). Veja a página 413 para obter um exemplo de como usá-lo.

Snow666

se esqueça de onde colocou sua distribuição de **sendmail**; normalmente, a localização dos arquivos é ditada pelo espaço em disco disponível, e não pela lógica de projeto do sistema de arquivos.

A macro OSTYPE

Arquivos no diretório **ostype** são nomeados para o sistema operacional cujos valores-padrão eles contêm. Um arquivo OSTYPE empacota uma série de informações específicas do fabricante, como os locais esperados para arquivos relacionadas com e-mail, caminhos para comandos de que **sendmail** precisa, flags para programas de entrega local etc.

Por convenção, as informações específicas a sistema operacional são interpoladas no arquivo de configuração com a macro OSTYPE.[29] Cada arquivo de configuração tem de incluir uma macro OSTYPE próximo do início, normalmente após `VERSIONID`.

Os arquivos `OSTYPE` fazem seu trabalho basicamente definindo outras variáveis **m4**. Por exemplo,

```
define(`ALIAS_FILE', `/etc/aliases')
```

especifica a localização do arquivo de nomes alternativos válidos para todo o sistema. Caso queira, você pode posteriormente fazer com que outros valores para o seu sistema operacional prevaleçam em relação aos valores-padrão no arquivo **.mc**, mas não altere o arquivo `OSTYPE` da distribuição original a menos que ele realmente esteja errado, em cujo caso você também deve abrir um relatório de bug e enviá-lo para sendmail-bugs@sendmail.org. Alguns sites querem uma localização consistente para o arquivo **aliases** entre plataformas e, portanto, redefinem sua localização em seus arquivos `DOMAIN`.

O arquivo **README** no diretório **cf** lista todas as variáveis que podem ser definidas num arquivo `OSTYPE`. Algumas das vaiáveis importantes são mostradas na Tabela 19.10, juntamente com várias que talvez você queira configurar para combate a spams (mas que, por padrão, não se encontram definidas). Os valores-padrão são o que você obtém caso seu arquivo `OSTYPE` não especifique nada em contrário.

Veja o Capítulo 18 para obter maiores informações sobre o NIS.

sendmail suporta o emprego de vários arquivos de nomes alternativos e mapas NIS, ambos para permitir o emprego simultâneo de arquivos e NIS e para facilitar a divisão de nomes alternativos entre arquivos globais e locais. Por exemplo,

```
define(`ALIAS_FILE', ``/etc/aliases,nis:mail.aliases'')
```

pesquisaria primeiramente o arquivo **/etc/aliases** e, caso não fosse bem-sucedido, tentaria o mapa NIS chamado mail.aliases.

Tabela 19.10 Valores-padrão de algumas variáveis configuradas nos arquivos OSTYPE.

Variável	Valor-padrão
`ALIAS_FILE`	**/etc/mail/aliases**
`HELP_FILE`	**/etc/mail/helpfile**
`STATUS_FILE`	**/etc/mail/statistics**
`QUEUE_DIR`	**/var/spool/mqueue**
`MSP_QUEUE_DIR`	/var/spool/clientmqueue
`LOCAL_MAILER_PATH`	**/bin/mail**
`LOCAL_SHELL_PATH`	**/bin/sh**
`LOCAL_MAILER_MAX`	*Indefinido*
`LOCAL_MAILER_MAXMSGS`	*Indefinido*
`SMTP_MAILER_MAX`	*Indefinido*
`SMTP_MAILER_MAXMSGS`	*Indefinido*

Se você instalar **sendmail** numa versão de sistema operacional nova ou numa nova arquitetura, certifique-se de criar um arquivo `OSTYPE` correspondente e dá-lo para sendmail.org para que possa ser incluído na próxima versão. Simplesmente modele o seu novo arquivo baseado naqueles que já existem e compare-o com a tabela de padrões no arquivo **cf/README**. Se o valor de uma variável em seu novo sistema for o mesmo que o valor-padrão, você não precisará incluir uma entrada para ele (mas não custa nada se proteger no caso do padrão mudar).

29. Portanto, onde afinal a macro OSTYPE em si é definida? Num arquivo no diretório **cf/m4** que é magicamente anexado ao seu arquivo de configuração ao você executar o script **Build**.

Snow666

A Tabela 19.11 ilustra os arquivos OSTYPE para as plataformas de referência adotadas neste livro. Somente o Red Hat usa o arquivo **linux.m4** da distribuição-padrão de **sendmail**.

Tabela 19.11 Arquivos OSTYPE para sistemas Linux.

Sistema	Arquivo	Diretório	Emprego
Red Hat	**linux.m4**	**/usr/share/sendmail-cf**	OSTYPE(`linux')
SuSE	**suse-linux.m4**	**/usr/share/sendmail**	OSTYPE(`suse-linux')
Debian	debian.m4	**/usr/share/sendmail/sendmail.cf**	OSTYPE('debian')

O SuSE coloca a distribuição de **sendmail** em **/usr/share/sendmail**. O arquivo OSTYPE **suse-linux.m4** se encontra no diretório **ostype** e não faz parte da distribuição **sendmail** de sendmail.org. Esse arquivo é muito longo (mais de 80 linhas) e contém numerosos recursos (FEATUREs) e outras macros que normalmente são encontradas no arquivo de configuração mestre de um site (o arquivo **.mc**), e não no arquivo OSTYPE. Isso oculta a real configuração do administrador de sistemas — uma bênção, quem sabe, mas *não* uma prática que nós recomendemos.

O Debian esconde os arquivos de configuração sob **/usr/share/sendmail/sendmail.cf/**. O diretório **sendmail.cf** (escolha confusa de nomes por parte do pessoal do Debian) corresponde ao diretório **cf** na distribuição **sendmail** e contém todas as peças de configuração de que você precisa, inclusive um arquivo OSTYPE específico ao Debian, **ostype/debian.m4**. O arquivo OSTYPE tem 50 linhas de comprimento e é formado em sua maior parte por nomes de caminhos e comentários, como deveria. Muitos são idênticos aos padrões atuais de sendmail.org e, portanto, realmente não precisam ser reiniciados explicitamente. Porém, reafirmá-los protege o Debian contra alterações nos padrões que de outra maneira poderiam introduzir inconsistências ou erros. O único pecado do Debian no que diz respeito ao OSTYPE de **sendmail** é a inclusão do arquivo DOMAIN genérico que vem junto com a distribuição **sendmail**. Uma instrução DOMAIN deve aparecer logo no início do arquivo .mc, e não ficar oculta no arquivo OSTYPE.[30]

A macro DOMAIN

A diretiva DOMAIN permite que informações genéricas válidas para todo o site sejam especificadas num só local (**cf/domain/***filename***.m4**) e depois sejam referidas no arquivo de configuração individual de cada host via

```
DOMAIN(`filename')
```

O nome de arquivo é normalmente escolhido para descrever o seu site. Por exemplo, nosso arquivo para o departamento de Ciências da Computação se chama **cs.m4** e aparece em nossos arquivos **.mc** como:

```
DOMAIN(`cs')
```

Assim como OSTYPE, DOMAIN é realmente apenas uma maneira interessante de fazer uma inclusão. Mas ele torna mais clara a estrutura do arquivo de configuração e fornece um gancho para futuros ajustes. Ele é útil principalmente quando centralizamos e construímos os arquivos **.cf** de todos os nossos sites por meio de arquivos **.mc** mantidos numa única localização.

Sites pequenos normalmente não precisam de um arquivo de domínio, mas sites maiores normalmente os utilizam para referências a máquinas retransmissoras, opções de privacidade ou de mascaramento válidas para todo o site e referências a tabelas para entregadores locais, domínios virtuais e bancos de dados de spam.

O arquivo DOMAIN genérico incluso com a distribuição mostra os tipos de entradas que normalmente são colocadas em arquivos de domínios válidos para todo o site. Seu conteúdo (sem comentários e dnls) é mostrado a seguir.

```
VERSIONID(`$Id: generic.m4,v 8.15 1999/04/04 00:51:09 ca Exp $')
define(`confFORWARD_PATH', `$z/.forward.$w+$h:$z/.forward+$h:
   $z/.forward.$w:$z/.forward')
define(`confMAX_HEADERS_LENGTH', `32768')
FEATURE(`redirect')
FEATURE(`use_cw_file')
EXPOSED_USER(`root')
```

30. A versão de sendmail 8.12.3 distribuída com a versão 3.0r1 do Debian coloca a instrução DOMAIN no arquivo **sendmail.mc** como deve ser (N.R.T.).

Snow666

Os arquivos configuram o caminho para as localizações dos arquivos de encaminhamento dos usuários, limita comprimentos de cabeçalho[31], incluem o recurso `redirect` para usuários que deixaram a empresa e ativam o recurso `use_cw_file` para a manipulação dos nomes de máquinas equivalentes. Se o seu arquivo **.mc** incluir mascaramento, o usuário root não será mascarado. Cada uma dessas construções é descrita de modo mais detalhado posteriormente, ainda neste capítulo.

A macro MAILER

Você tem de incluir uma macro `MAILER` para cada agente de entrega que quiser habilitar. Você encontrará uma lista completa dos agentes de entrega suportados no diretório **cf/mailers** na distribuição **sendmail**. Atualmente, as opções são `localf`, `smtp`, `fax`, `usenet`, `procmail`, `qpage`, `cyrus`, `pop`, `phquery` e `uucp`. Tipicamente, você precisará pelo menos de:

```
MAILER(`local')
MAILER(`smtp')
```

A primeira linha inclui os agentes de entrega `local` e `prog`. A segunda linha inclui `smtp`, `esmtp`, `dsmtp`, `smtp8` e `relay`. Foi acrescentado suporte a endereços de e-mail *usuario+detalhes@site.dominio* ao agente de entrega local a partir da versão 8.7.[32] *usuario* define a caixa postal para a qual as mensagens devem ser entregues e *detalhes* fornece um parâmetro extra que um programa de e-mail local como **procmail** poderá usar para ordenar e-mails que chegam.

Caso planeje ajustar qualquer macro relacionada a agentes de entrega (como `USENET_MAILER_ARGS` ou `FAX_MAILER_PATH`), certifique-se de que as linhas que configuram estes parâmetros *precedam* a linha que chama o agente de entrega em si; caso contrário, os valores antigos serão usados. Por essa razão, as declarações `MAILER` normalmente vêm mais para o final do arquivo de configuração.

O agente de entrega `pop` faz a interface para o programa **spop** que faz parte do pacote manipulador de e-mails **mh** e implementa o Post Office Protocol definido na RFC1460. Ele é usado por PCs e Macs que precisam acessar e-mails num host UNIX. O agente de entrega `cyrus` é para uso com o servidor IMAP da CMU e vem em duas variantes: `cyrus` para entregar e-mails para caixas postais de usuários e `cyrusbb` para entregar e-mails para um sistema eletrônico de quadros de mensagens centralizado. O agente de entrega `cyrus` também compreende a sintaxe usuario+detalhes; sua especificação `MAILER` tem que vir depois daquela do agente de entrega `local`.

O agente de entrega `fax` integra o pacote HylaFAX de Sam Leffler no sistema de e-mail. O SuSE o inclui como **/usr/bin/faxmail**; o Red Hat e o Debian não incluem o HylaFAX na situação-padrão.

HylaFAX se encontra disponível em www.hylafax.org.

Enviar e-mails para *usuario@destino*.fax envia o corpo da mensagem como um documento de fax. *destino* é, tipicamente, um número de telefone. Para possibilitar que nomes simbólicos (em vez de apenas números de telefones) sejam utilizados como destinos, use um arquivo de banco de dados com chaves.

***ghostscript** se encontra disponível em www.gnu.org.*

Você tem de juntar HylaFAX a **sendmail** instalando um script da distribuição HylaFAX em **/usr/local/bin**. Talvez você também precise alterar o valor da macro `FAX_MAILER_PATH`. Intervenção humana ainda é necessária para transferir faxes que chegam da área de spool para a caixa postal de um usuário. Você pode converter documentos fax para PostScript (com HylaFAX) e visualizá-los com o pacote GNU **ghostscript**.

O agente de entrega `qpage` faz a interface para o software QuickPage para entregar e-mails para o seu pager. Vá ao site www.qpage.org para obter maiores informações sobre o QuickPage.

As macros `VERSIONID`, `OSTYPE` e `MAILER` são tudo o que você precisa para construir um arquivo *hostname*.**mc** básico.

19.8 Primitivas de Configuração para Sendmail Mais Incrementadas

Nas próximas seções, descrevemos algumas outras macros e alguns dos recursos (`FEATURE`s) mais comuns usados para modificar o comportamento-padrão de **sendmail**. Também discutimos algumas questões de política no contexto da configuração de **sendmail**: segurança, privacidade, spam e a técnica de ocultar informações pelo emprego de mascaramento e de domínios virtuais.

A macro FEATURE

Através da macro `FEATURE` podemos habilitar várias opções comuns incluindo arquivos **m4** do diretório **feature**. Na discussão a seguir, entremeamos nossa apresentação de `FEATURE`s e algumas outras macros de **sendmail**, já que oca-

31. Os hackers usaram cabeçalhos extremamente longos como uma forma de provocar uma recusa de serviços em versões mais antigas de **sendmail**. Essa linha está aí no caso de você ainda usar uma dessas versões vulneráveis (anteriores à versão 8.9.3).
32. A sintaxe usuario+detalhes originou-se na Carnegie Mellon University, onde ele é usado com ferramentas locais para direcionar e ordenar e-mails.

www.snow666.blogspot.com
Snow666

sionalmente elas se entrelaçam. Quando a configuração **m4** foi acrescentada pela primeira vez a **sendmail**, descrever a macro `FEATURE` se tornou uma grande seção de nosso capítulo sobre e-mail. Agora, foram adicionados tantos recursos que `FEATURE` quase precisa de um capítulo próprio. A sintaxe é:

```
FEATURE(palavrachave, arg, arg, _)
```

onde *palavrachave* corresponde a um arquivo *palavrachave*.**m4** no diretório **cf/feature**, e os argumentos (*args*) são passados para ele. Veja o próprio diretório ou o arquivo **cf/README** para uma lista conclusiva dos recursos. Alguns poucos mais comumente usados são descritos a seguir.

O recurso use_cw_file

A classe `w` interna de **sendmail** (daí o nome **cw**) contém os nomes de todos os hosts locais que este host aceita e entrega e-mails. Uma máquina-cliente poderia incluir o seu nome de host, seus apelidos e localhost nesta classe. Se o host que está sendo configurado for o seu centralizador de e-mail, então a classe `w` também deve incluir quaisquer hosts locais e domínios virtuais para os quais você aceita e-mails.

O recurso `use_cw_file` define a classe `w` por meio do arquivo **/etc/mail/local-host-names** (que costumava ser chamado de **sendmail.cw**). O nome de arquivo exato é configurável pela opção `confCW_FILE`, discutida posteriormente. Sem esse recurso, **sendmail** entrega e-mails localmente somente se forem endereçados à máquina na qual **sendmail** está sendo executado. Um servidor para e-mails que chegam deve listar no arquivo **local-host-names** todos os nomes de domínios e máquinas para os quais ele irá manipular e-mails. Se você alterar o arquivo, terá de enviar um sinal HUP a **sendmail** para que suas alterações entrem em efeito, pois **sendmail** lê este arquivo somente ao iniciar.

```
FEATURE(`use_cw_file')
```

chama o recurso e usa o arquivo **local-host-names** como fonte de dados; eis um exemplo de arquivo **local-host-names**:

```
# local-host-names - inclui todos os nomes alternativos para a sua máquina neste ponto.
toadranch.com
coyote.toadranch.com
hein.tv
coyote.hein.tv
big-tr.com
yoherb.com
herbmorreale.com
appliedtrust.com
applied-trust.com
atrust.com
```

Nesse exemplo, as entradas são domínios virtuais hospedados localmente.

O recurso redirect

Quando funcionários deixam uma empresa, normalmente encaminhamos seus e-mails ou deixamos que eles ricocheteiem de volta para o remetente com um erro. O recurso `redirect` fornece suporte para uma maneira mais elegante de ricochetear e-mails. Se Joe Smith se formou e passou de oldsite.edu para newsite.com, então habilitar o `redirect com`

```
FEATURE(`redirect')
```

e acrescentar a linha

```
smithj: joe@newsite.com.REDIRECT
```

ao arquivo **aliases** faz com que e-mails para smithj sejam retornados para o remetente com uma mensagem de erro que sugere que o remetente tente o endereço joe@newsite.com em seu lugar. A mensagem em si não é enviada automaticamente.

O recurso always_add_domain

Este recurso faz com que **sendmail** adicione o nome de host local aos endereços de destino locais que não sejam totalmente qualificados. Suponha, por exemplo, que lynda@cs.colorado.edu envie uma mensagem para os usuários locais barb e evi. Sem `always_add_domain`, os cabeçalhos de e-mails exibiriam os endereços do remetente e do receptor como nomes de login simples. Com o recurso `always_add_domain` ativado, todos os endereços se tornarão totalmente qualificados antes de a mensagem deixar a máquina de lynda.

www.snow666.blogspot.com
Snow666

Use `always_add_domain` ao compartilhar diretórios de spool entre máquinas que não compartilhem um arquivo de nome alternativo ou que não tenham o mesmo arquivo **passwd** (incidentemente, com certeza você não deve fazer tal compartilhamento). Os e-mails para um nome alternativo ou usuário que não é conhecido em lugar nenhum devem ser totalmente qualificados na máquina da qual eles se originam e, portanto, poderão ser respondidos.

Um outro ponto a favor para este recurso é que nomes não qualificados normalmente são rejeitados como spam. Recomendamos que você sempre o utiliza (a menos que esteja enviando spam!).

Se estiver usando `MASQUERADE_AS` (veja a seção 19.8), `always_add_domain` acrescenta o nome do host que você está se fazendo passar, não o nome de host local. Essa convenção pode provocar problemas se o arquivo **aliases** ou o arquivo **passwd** no host local não for um subconjunto do arquivo equivalente no host `MASQUERADE_AS`.

O recurso nocanonify

sendmail tipicamente verifica se a parte de nome de domínio de um endereço é totalmente qualificada e não um CNAME no DNS. Se isso não for verdade, **sendmail** reescreve o endereço. Esse processo se chama canonificação e, normalmente, é feito executando-se uma pesquisa DNS no nome de host. O recurso `nocanonify` diz não a essa reescrita, e a pesquisa DNS que é necessária para entregar uma mensagem é postergada. Por exemplo, num site com um centralizador de e-mail mestre e máquinas-clientes que encaminham todos os seus e-mails através do mestre, os clientes poderiam usar

```
FEATURE(`nocanonify')
```

para impedir a realização de pesquisas DNS localmente. **sendmail** não controla que pesquisas DNS foram feitas à medida que uma mensagem se desloca de máquina em máquina dentro de um site local — ele não é capaz de fazê-lo. O recurso `nocanonify` permite que você controle o período dessas pesquisas. Veja o nosso segundo modelo de configuração (página 430) para um exemplo.

`nocanonify` também pode ser usado num esquema MSA/MTA como o que poderia ser utilizado num grande site de e-mails. Nessa situação, o MSA realiza todas as pesquisas DNS, e a máquina mestra que executa o MTA especifica `nocanonify`.

Algumas vezes queremos impedir pesquisas DNS que são potencialmente caras, mas queremos realizar as pesquisas para o domínio local. Você pode dispensar domínios específicos da especificação `nocanonify` incluindo tanto as macros `CANONIFY_DOMAIN` ou `CANONIFY_DOMAIN_FILE`, que usam, respectivamente, uma lista de domínios ou um nome de arquivo como argumento. Por exemplo, as linhas

```
FEATURE('nocanonify')
CANONIFY_DOMAIN('cs.colorado.edu cs')
```

postergariam pesquisas DNS, exceto para endereços no formato *usuario*@cs.colorado.edu ou *usuario*@cs. Essas macros de exceção foram introduzidas pela primeira vez na versão 8.12.

Tabelas e bancos de dados

sendmail possui vários recursos (`FEATURE`s) que usam uma construção chamada de "tabela" para descobrir para onde o e-mail deve ser direcionado. Uma tabela normalmente é um arquivo de texto de roteamento, nomes alternativos, política ou outras informações, convertidas externamente para um formato de banco de dados com o comando **makemap** e depois usadas como banco de dados interno para diversas pesquisas de **sendmail**. Embora os dados normalmente comecem como um arquivo de texto, isto não é obrigatório; os dados para as tabelas **sendmail** podem provir do DNS, NIS, LDAP ou outras fontes. O emprego de um servidor IMAP ou POP centralizado libera **sendmail** da tarefa de caçar usuários e tornam obsoletas algumas das tabelas discutidas a seguir. A Tabela 19.12 sintetiza as tabelas disponíveis.

São suportadas duas bibliotecas de bancos de dados: a biblioteca **dbm/ndbm** que é o padrão na maioria das versões do Linux; e Berkeley DB, uma biblioteca mais extensível que suporta vários esquemas de armazenamento. Sua escolha em termos de bibliotecas de bancos de dados tem de ser especificada no momento da compilação. Recomendo DB caso você possa instalá-la; ela é mais rápida do que **dbm** e cria arquivos menores. DB se encontra disponível em sleepycat.com.

Há três tipos de mapas de bancos de dados disponíveis:

- `dbm` — usa um algoritmo de hashing extensível (**dbm/ndbm**)
- `hash` — usa um esquema de hashing padrão (DB)
- `btree` — usa uma estrutura de dados árvore-B (DB)

Snow666

Para a maioria dos aplicações de tabelas no **sendmail**, o tipo de banco de dados `hash` — o padrão — é o melhor. Use o comando **makemap** para construir o arquivo de banco de dados por meio de um arquivo de texto; você especifica o tipo de banco de dados e o nome-base do arquivo de saída. A versão em forma de texto do banco de dados deve aparecer na entrada-padrão de **makemap**, por exemplo:

```
# makemap hash /etc/mail/access < /etc/mail/access
```

À primeira vista esse comando se parece com um erro que faria com que o arquivo de entrada fosse sobregravado por um arquivo de saída vazio. Entretanto, **makemap** agrega um sufixo apropriado, de modo que o arquivo de saída real é **/etc/mail/access.db** e, na realidade, não há nenhum conflito. Cada vez que o arquivo de texto for modificado, o arquivo de banco de dados terá de ser reconstruído com **makemap** (mas não é necessário enviar o sinal HUP para **sendmail**).

Na maioria das circunstâncias é usada a mais longa concordância possível para chaves de bancos de dados. Como acontece com qualquer estrutura de dados hashed, a ordem das entradas no arquivo de texto de entrada não é significativa. `FEATURE`s que esperam um arquivo de banco de dados como parâmetro usam `hash` como padrão para o tipo de banco de dados e **/etc/mail/***nomedatabela***.db** como o nome de arquivo para o banco de dados. Para prevalecer sobre esse comportamento, especifique o tipo de banco de dados desejado tanto para o comando **makemap** quanto para o `FEATURE` ou reinicialize o padrão definindo um valor diferente para a variável `DATABASE_MAP_TYPE`. Por exemplo:

```
define(`DATABASE_MAP_TYPE', `dbm')
```

Para usar o seu novo banco de dados **access.db**, você adicionará a seguinte linha ao seu arquivo **.mc**:

```
FEATURE(`access_db', `hash /etc/mail/access')
```

Já que essa linha usa o tipo e o esquema de atribuição de nomes-padrão, você poderia simplesmente escrever

```
FEATURE(`access_db')
```

Você pode especificar o nome de arquivo de banco de dados seja com ou sem o sufixo (**.db**); sem é preferível.

Não se esqueça de reconstruir o arquivo de banco de dados com **makemap** toda vez que alterar o arquivo de texto, ou suas alterações não terão efeito.

Abordamos os recursos `mailertable`, `genericstable` e `virtusertable` nas próximas seções. `access_db` é visto posteriormente na seção de spam. `user_db` não é visto, pois, eventualmente, será eliminado.

A partir da versão 8.12, todos os mapas e classes podem especificar LDAP como fonte de seus dados, de modo que você pode solicitar que **sendmail** contacte o servidor LDAP para determinar o direcionamento do e-mail e o novo cabeçalho. Apenas especifique `LDAP como segundo` parâmetro:

```
FEATURE(`access_db', `LDAP')
```

Essa linha fará com que `access_db` use o esquema LDAP padrão que é definido no arquivo **cf/sendmail.schema** na distribuição **sendmail**. Você também pode definir o seu próprio esquema de banco de dados com argumentos adicionais à diretiva `FEATURE`; veja o arquivo **cf/README** para maiores detalhes.

O recurso mailertable

O recurso `mailertable` redireciona e-mails endereçados a um determinado host ou domínio para um destino alternativo através de um agente de entrega em particular. Ele é aplicado quando o e-mail sai de um site. O recurso `mailertable` pesquisa somente a porção host do endereço, e não a porção usuário. O endereço de cabeçalho não é reescrito, de modo que o e-mail continua a ser endereçado ao mesmo usuário, porém ele é enviado a um host diverso por meio de um agente de entrega diferente. `mailertable` foi concebido originalmente para lidar com outros sistemas de e-mail como UUCP, DECnet e BITNET, mas hoje é usado para redirecionar e-mails de uma máquina gateway para um servidor interno.

Para usar um `mailertable`, inclua a seguinte linha em seu arquivo **.mc**.

```
FEATURE(`mailertable')
```

www.snow666.blogspot.com
Snow666

Uma entrada em mailertable possui o seguinte formato:

```
dominio_antigo        mailer:destino
```

Um ponto em frente da chave no lado esquerdo é um caractere-curinga significando qualquer host neste domínio. Somente nomes de domínio e host são permitidos como chaves `mailertable`; nomes de usuários não são permitidos. O valor *destino* do lado direito pode ser um domínio, uma estrutura usuario@dominio ou, até mesmo, nulo, em cujo caso o envelope não é alterado. O valor *mailer* tem de ser o nome de um agente de entrega definido numa estrutura `MAILER`; veja mais à frente, nesta seção .

Suponha, por exemplo, que você usasse MS Exchange como o seu servidor de e-mails interno principal, porém fosse relutante em permitir que ele mantivesse contato com a Internet. Você colocaria um computador Linux na Internet como o seu gateway de e-mails e a seguir encaminharia todos os e-mails para o servidor Exchange depois de ter sido feita uma varredura de vírus ou seja lá qual for o pré-processamento desejado. Eis a entrada `mailertable` que faria isso, supondo-se que o servidor Exchange tivesse o endereço IP interno exibido:

```
meu-dominio        esmtp:[192.168.1.245]
```

Entretanto, essa é uma forma de retransmissão que, como será visto na seção 19.10, precisa ser controlada. Para completar esse exemplo, você precisaria colocar a linha

```
To: meu-dominio RELAY
```

em seu banco de dados de acesso para permitir retransmissão para todos os e-mails dirigidos a qualquer usuário em meu-dominio.

O recurso genericstable

O recurso `genericstable` ("generics table" e não "generic stable", ou seja, "tabela genérica" e não " genérico estável") é como nome alternativo para e-mails de saída. Ele pode, por exemplo, associar trent@xor.com a trent.hein@xor.com nos e-mails que saem. São os cabeçalhos escritos novamente, não o envelope. A entrega de e-mails não é afetada, somente as respostas.

Vários mecanismos podem ser usados para mapear nomes de hosts, mas `genericstable` é o único recurso que inclui tanto o nome de usuário quanto o nome de host na chave de mapeamento. Os recursos `masquerade_envelope` e `allmasquerade` discutidos posteriormente nesta seção também podem ser aplicados a endereços no recurso `genericstable`.

Para usar `genericstable`, certifique-se de que o seu domínio se encontre na classe `generics`. Para colocar um domínio na classe `generics`, você pode listá-lo na macro `GENERICS_DOMAIN` ou colocá-lo no arquivo especificado pela macro `GENERICS_DOMAIN_FILE`.

Por exemplo, para usar `genericstable` com os padrões para o banco de dados, acrescente

```
GENERICS_DOMAIN_FILE(`/etc/mail/local-host-names')
FEATURE(`genericstable')
```

ao seu arquivo de configuração **.mc**. Nesse exemplo, qualquer host para o qual você aceite e-mails será incluído. Habilitar o recurso `genericstable` diminui um pouco o desempenho de **sendmail**, pois todos os endereços de remetente têm de ser pesquisados.

O recurso virtusertable

A tabela de usuários virtual suporta nomes alternativos de domínios para e-mails que chegam. Este recurso permite que vários domínios virtuais sejam hospedados em uma máquina e é comum em sites de hospedagem Web.

O campo key da tabela contém um endereço de e-mail (*usuario@host.dominio*) ou uma especificação de domínio (*@dominio*). O campo value é um endereço de e-mail local ou um endereço de e-mail externo. Se key for um domínio, o valor pode passar o campo *usuario* juntamente com a variável `%1` ou direcionar o e-mail para um usuário diferente. Se a especificação de usuário tiver a forma *usuario+detalhes*, então a variável `%2` conterá *detalhes* e variável `%3` conterá *+detalhes*; use seja lá qual for a forma que você desejar.

Vejamos alguns exemplos (nós acrescentamos os comentários):

```
info@foo.com    foo-info              # direciona para um usuário local
info@bar.com    bar-info              # um outro usuário local
joe@bar.com     error:No such user    # para retornar um erro
@baz.org        jane@elsewhere.com    # todos os e-mails para jane
@zokni.org      %1@elsewhere.com      # para o mesmo usuário, domínio diferente
```

Snow666

Todas as chaves de host do lado esquerdo dos mapeamentos de dados têm de ser listadas no arquivo **cw**, **/etc/mail/local-host-names** (ou a classe `VirtHost`); caso contrário, **sendmail** tenta localizar o host na Internet e entregar o e-mail lá. Se DNS apontar **sendmail** de volta para esse servidor, você obterá uma mensagem "local configuration error — erro de configuração local" em ricochetes. Infelizmente, **sendmail** não é capaz de dizer que a mensagem de erro para essa instância deverá realmente ser "chave virtusertable não se encontra no arquivo cw".

Várias peças estão realmente envolvidas aqui:

- Registros MX de DNS têm de existir, de modo que e-mails são direcionados para o host correto em primeiro lugar, depois
- entradas **cw** têm de estar presentes ou `VIRTUSER_DOMAIN` especificada (ou de maneira equivalente, `VIRTUSER_DOMAIN_FILE`) para permitir que a máquina local aceite o e-mail e, finalmente
- a tabela de usuário virtual tem de informar **sendmail** o que fazer com o e-mail.

O recurso é chamado com:

```
FEATURE(`virtusertable')
```

Os exemplos da seção 19.9 usam `virtusertable` para implementar hospedagem virtual.

O recurso ldap_routing

E como último participante neste antro de nomes alternativos, reescrita e falsificação, temos o LDAP (Lightweight Directory Access Protocol). LDAP (veja a página 403 para informações genéricas) pode ser usado como substituto para o recurso `virtusertable` em relação ao direcionamento de e-mails e aceitação de e-mails para domínios virtuais. Ele também pode gerenciar nomes alternativos, mapas e classes. E, a partir da versão 8.12, ele pode realizar um trabalho decente com listas de discussão.

Para usar LDAP dessa maneira, você tem de incluir várias instruções em seu arquivo de configuração e tem de construir **sendmail** para incluir suporte ao LDAP. Em seu arquivo **.mc** você precisa das linhas

```
define(`confLDAP_DEFAULT_SPEC', `-h servidor -b basedeprocura')
FEATURE(`ldap_routing')
LDAPROUTE_DOMAIN(`meu-dominio')
```

para informar a **sendmail** que você quer usar um bd LDAP para direcionar e-mails que chegam endereçados ao domínio especificado. A opção `LDAP_DEFAULT_SPEC` identifica o servidor LDAP e o nome-base de pesquisa de banco de dados.

No exemplo a seguir, a base de pesquisa é `o=sendmail.com, c=US`. Se você estiver usando LDAP numa porta personalizada (não 389), adicione `-p` *ldap_port#* para `LDAP_DEFAULT_SPEC`.

sendmail usa os valores de duas tags no banco de dados LDAP:

- `mailLocalAddress` para o destinatário de e-mails que chegam
- `mailRoutingAddress` para o nome alternativo ao qual enviar

sendmail também suporta a tag `mailHost`, que, se presente, direciona e-mails para os registros MX para o host especificado, com o recurso `mailRoutingAddress` como receptor.

Por exemplo, a entrada LDAP (para um servidor configurado com um nome root distinto `cn=root, o=sendmail.com, c=US`)

```
dn: uid=eric, o=sendmail.com, c=US
objectClass: inetLocalMailRecipient
mailLocalAddress: eric@sendmail.org
mailRoutingAddress: eric@eng.sendmail.com
```

faria com que e-mails endereçados para eric@sendmail.org (que os registros MX do DNS fizeram com que fossem entregues para sendmail.com) fossem enviados para eric@eng.sendmail.com. Se a entrada também contivesse a linha

```
mailHost: mailserver.sendmail.com
```

então, e-mails para eric@sendmail.org seriam endereçados para eric@eng.sendmail.com e enviados para o host mailserver.sendmail.com após pesquisas MX.

Snow666

As entradas de banco de dados LDAP suportam uma entrada de caractere curinga, *@dominio*, que redireciona os e-mails endereçados a qualquer um no domínio especificado (como foi feito no recurso `virtusertable`).

Nas versões 8.12 e posteriores, foi acrescentado um pouco mais de flexibilidade na forma de uma primitiva de configuração, `LDAPROUTE_EQUIVALENT` (ou `LDAPROUTE_EQUIVALENT_FILE`), que permite definir versões equivalentes do nome de domínio que estamos redirecionando com LDAP. Por exemplo, e-mails chegando para usuario@host1.meudominio normalmente seriam consultados literalmente no banco de dados LDAP e depois consultados como @host1.meudominio. Incluindo-se a linha

```
LDAPROUTE_EQUIVALENT(`host1.meudominio')
```

também seriam tentadas as chaves usuario@meudominio e @meudominio. Esse recurso permite que seja usado um único banco de dados para direcionar e-mails num site complexo.

Argumentos adicionais ao recurso `ldap_routing` agora permitem que especifiquemos mais detalhes sobre o esquema LDAP para usar e especificar a manipulação de nomes de usuário que possuem uma parte +detalhe. Como sempre, veja o arquivo **cf/README** para obter maiores informações.

Mascaramento e a macro MASQUERADE_AS

Com a macro `MASQUERADE_AS`, podemos especificar uma única identidade atrás da qual outras máquinas se ocultam. Todos os e-mails parecem emanar da máquina ou domínio designado. O endereço do remetente é reescrito para ser *usuario@nome-mascaramento* em vez de *usuario@host-original.dominio*. Obviamente, esses endereços mascarados têm de ser válidos para que as pessoas possam responder aos e-mails.

Essa configuração permite que todos os usuários de um site usem um endereço de e-mail genérico. Por exemplo, se todos os hosts em xor.com se disfarçarem por trás do domínio xor.com, então e-mails de *usuario@host*.xor.com serão estampados como provenientes de *usuario*@xor.com, sem nenhuma menção do nome de host verdadeiro por meio do qual o usuário enviou o e-mail. A máquina que representa xor.com tem de saber como entregar os e-mails de todos os usuários, mesmo e-mails para usuários que não tenham um login no servidor de e-mails de entrada. Naturalmente, nomes de login devem ser exclusivos através de todo o domínio.

Alguns usuários e endereços (como root, postmaster, hostmaster, trouble, operations, Mailer-Daemon etc.) devem ser excluídos desse comportamento. Eles podem ser excluídos explicitamente com a macro `EXPOSED_USER`. Por exemplo, a seqüência

```
MASQUERADE_AS(`xor.com')
EXPOSED_USER(`root')
EXPOSED_USER(`Mailer-Daemon')
```

estamparia e-mails como provenientes de usuario@xor.com a menos que tenham sido enviados por root ou pelo sistema de e-mails; nestes casos, os e-mails transportariam o nome do host de origem.

Um recurso introduzido na versão 8.12 permite que você exclua os e-mails para o domínio local (ou e-mails para os hosts específicos listados como exceções) do mascaramento. Por exemplo, esse recurso poderia ser útil para um site que use um nome de domínio privado não registrado localmente e desejasse mascaramento somente em mensagens ligadas à Internet.

A sintaxe é:

```
FEATURE('local_no_masquerade')
MASQUERADE_EXCEPTION(`host.dominio')
MASQUERADE_EXCEPTION_FILE(`nomedearquivo')
```

Há várias extensões para a macro `MASQUERADE_AS` básica, ambas por meio de outras macros e de recursos (`FEATURE`s):

- A macro `MASQUERADE_DOMAIN`
- A macro `MASQUERADE_DOMAIN_FILE`
- A macro `MASQUERADE_EXCEPTION`
- A macro `MASQUERADE_EXCEPTION_FILE`
- O recurso `limited_masquerade`
- O recurso `allmasquerade`
- O recurso `masquerade_envelope`
- O recurso `masquerade_entire_domain`

www.snow666.blogspot.com
Snow666

Recomendamos usar a macro MASQUERADE_AS juntamente com os recursos allmasquerade e masquerade_envelope. O recurso limited_masquerade modifica o comportamento de MASQUERADE_DOMAIN e é útil para ambientes de hospedagem virtual. MASQUERADE_DOMAIN permite que você liste domínios que deseja mascarar; a lista é pré-carregada por meio da classe w que é tipicamente definida com o recurso use_cw_file e lista os hosts em seu domínio. limited_masquerade não pré-inicializa a lista com a classe w. Todos esses domínios serão ocultos pelo domínio com o qual você está mascarando.

O recurso allmasquerade estende o mascaramento para os receptores da mensagem (ao contrário de apenas o remetente), e o recurso masquerade_envelope o estende ao envelope, bem como os endereços de cabeçalho.[33] Com essas duas extensões, todos os endereços são ocultos de uma forma consistente. O recurso masquerade_entire_domain estende o mascaramento a todos os hosts numa lista especificada de outros domínios.

Se quiser usar outras técnicas de mascaramento, poderá ler sobre seus comportamentos no arquivo **cf/README** ou no livro sobre **sendmail**. Leia com atenção; algumas das primitivas de mascaramento podem ocultar muita coisa.

As macros MAIL_HUB e SMART_HOST

O mascaramento faz com que todos os e-mails pareçam ser provenientes de um único host ou domínio, reescrevendo os cabeçalhos e, opcionalmente, o envelope. Alguns sites podem querer que todos os e-mails realmente provenham (ou sejam enviados para) uma única máquina. Você pode obter essa configuração com as macros MAIL_HUB para e-mails que chegam e SMART_HOST para e-mails que saem.

Se você quiser redirecionar todos os e-mails que chegam a um servidor central para entrega, configure MAIL_HUB para o valor *mailer:host*, onde *mailer* é o agente a ser usado para atingir o *host* designado. Se nenhum agente de entrega for especificado, relay é usado. Por exemplo:

```
define(`MAIL_HUB', `smtp:mailhub.cs.colorado.edu')
```

A designação SMART_HOST faz com que um host entregue e-mail local, mas mande e-mails externos para SMART_HOST. Esse recurso é útil para máquinas por trás de um firewall que não podem usar o DNS diretamente. Essa sintaxe se compara àquela de MAIL_HUB; o agente de entrega padrão é novamente relay. Por exemplo:

```
define(`SMART_HOST', `smtp:mailhub.cs.colorado.edu')
```

Nesses exemplos, a mesma máquina atua como servidor tanto para e-mails que chegam quanto para aqueles que saem. Um site maior poderia dividir isso em máquinas separadas. O SMART_HOST tem de permitir a retransmissão para que as máquinas-cliente possam enviar e-mails através dele. Entradas mailertable prevalecem sobre a designação SMART_HOST.

Mascaramento e roteamento

Com todos esses recursos e macros prontos e esperando para manipular seus endereços de e-mail, imaginamos que seria interessante tentar comparar os diversos mecanismos em termos de alterarem os cabeçalhos, o envelope ou a entrega de uma mensagem, ou se eles se aplicam a mensagens que chegam e que saem, endereços de remetente ou de receptor etc. Se a página tivesse uma largura dupla ou tripla, talvez pudéssemos realmente ter sido bem-sucedidos em ilustrar as diferenças entre os diversos comandos.

Em vez disso, simplesmente lhe damos uma dica; você terá de pesquisar os detalhes na documentação de **sendmail** para obter todas as nuances das diversas variações.

As entradas da Tabela 19.12 com todas as letras em maiúsculo são macros **m4**. As entradas com letras minúsculas são os nomes dos recursos que foram chamados com a macro FEATURE. Itens recuados dependem dos itens acima; por exemplo, um recurso que modifica o comportamento MASQUERADE_AS não faz nada a menos que MASQUERADE_AS tenha sido ativado. Na tabela, o recurso seria recuado para indicar essa dependência. O mascaramento afeta os endereços de cabeçalho em e-mails que saem e se uma mensagem pode ser respondida; o roteamento afeta a entrega em si dos e-mails.

---

33. Os endereços de cabeçalho são os endereços To, From, Cc e Bcc que aparecem no cabeçalho de uma mensgem. Os endereços de envelope são os endereços para o qual o e-mail é efetivamente entregue. Os endereços de envelope são originalmente construídos por meio dos endereços de cabeçalho pelo agente de usuário, mas eles são processados separadamente por **sendmail**. Muitos dos recursos de mascaramento e de redirecionamento de **sendmail** seriam impossíveis de serem implementados caso a distinção entre os endereços de cabeçalho e de envolope não fosse mantida.

Snow666

O recurso nullclient

`nullclient` é usado para um host que jamais deveria receber e-mails diretamente e que envie todos os seus e-mails para um servidor central. O arquivo **.mc** para um host desses possui somente duas linhas.

```
OSTYPE(`ostype')
FEATURE(`nullclient', `servidor-de-email')
```

O recurso `nullclient` prevalece sobre muitos outros recursos. Todos os e-mails, sem exceção, são entregues para *servidor-de-email* para processamento.[34] Note que o servidor tem de permitir que o cliente retransmita através dele caso os usuários regularmente originem e-mails no cliente e não usem um servidor distinto para e-mails que saem. Versões recentes de **sendmail** têm a retransmissão desativada como padrão. Veja a seção 19.10, sobre spam, para maiores detalhes sobre como controlar a retransmissão. Uma configuração `nullclient` se mascara como *servidor-de-email*, de modo que talvez você queira incluir uma instrução `EXPOSED_USER` para root.

Tabela 19.12 Comparação entre os recursos de mascaramento e de roteamento.

	Construção	Direção	Afeta[a]	Que trecho
MASCARAMENTO	MASQUERADE_AS	para fora	SH	host.dominio
	allmasquerade	para fora	RH[b]	host.dominio
	MASQUERADE_DOMAIN[_FILE]	para fora	SH	host.dominio
	masquerade_entire_domain	para fora	SH	host.sub.dominio
	limited_masquerade	para fora	SH	host.dominio
	masquerade_envelope	para fora	SE[c]	host.dominio
	genericstable	para fora	SH	user@host.dominio
ROTEAMENTO	`mailertable`	para fora	MAD	host.dominio
	`virtusertable`	para dentro	RD	user@host.dominio
	`ldap`	para dentro	RD	user@host.dominio
	`mailhub`	para dentro	RD	e-mail local
	`smarthost`	para fora	RD	e-mail remoto

a. S = sender (remetente), R = recipient (receptor), D = delivery (entrega), H = header (cabeçalho), E = envelope, M = mailer (agente de entrga), A = address (endereço).
b. Assim que a reescrita de receptor tiver sido habilitada através do recurso `allmasquerade`, todas as demais construções de mascaramento reescrevem não somente o remetente, como também o receptor.
c. Assim que a reescrita de envelope tiver sido habilitada através do recurso `masquerade_envelope`, todas as demais construções de mascaramento reescrevem não somente o cabeçalho, como também o envelope.

O cliente que usa o recurso `nullclient` deve ter um registro MX associado que aponte para o servidor. Ele também tem de ser incluído no arquivo **cw** do servidor, que normalmente é **/etc/mail/local-host-names**. Esses ajustes de configuração permitem que o servidor aceite e-mails para o cliente.

Um host com uma configuração `nullclient` não deve aceitar e-mail que chega. Se fizer isso, ele simplesmente encaminhará o e-mail para o servidor de qualquer maneira. Iniciar **sendmail** sem o flag **-bd** de modo que ele não escute conexões SMTP na porta 25 é uma maneira de impedir o recebimento de e-mails. Entretanto, alguns agentes de usuário (MUAs) tentam a submissão inicial de uma mensagem de e-mail através da porta 25 e, portanto, serão frustrados em seu intento caso seu **sendmail** não esteja escutando. Uma maneira melhor para desabilitar a chegada de e-mails é executar **sendmail** com o flag **-bd**, mas usar DAEMON_OPTIONS para escutar somente na interface de loopback. De qualquer maneira, deixe o flag **-q30m** na linha de comando; se o *servidor-de-email* cair, o cliente poderá colocar em fila o seu e-mail e tentar enviá-lo mais tarde para *servidor-de-email*.

`nullclient` é apropriado para nós de extremidade em sites que possuem uma máquina central para e-mails. Em sites maiores, considere a carga de e-mails na máquina centralizadora. Talvez seja interessante separar os servidores de recebimento e de envio ou adotar uma abordagem hierárquica.

34. Se você configurar um cliente dessa maneira e depois testar a configuração com **sendmail -bt**, o cliente parecerá entregar localmente e-mails locais. A razão é que a diretiva `nullclient` é processada posteriormente, em ruleset 5 do arquivo de configuração bruto.

Snow666

O Linux SuSE vem com um arquivo **mc** `nullclient` de modelo no diretório **/etc/mail/linux.nullclient.mc**. Basta preencher o nome de *servidor-de-email*, executar **m4** nele para construir o arquivo **sendmail.cf**, e pronto.

Os recursos local_lmtp e smrsh

Como padrão, o agente de entrega `local` usa **/bin/mail** como agente de entrega local para usuários e arquivos, e **/bin/sh** como agente de entrega para programas. **sendmail** agora dispõe de alternativas melhores, especialmente para a entrega para programas. Ambos estão disponíveis com a macro `feature`.

Se o recurso `local_lmtp` for especificado, então seu argumento é um agente de entrega local capaz de falar LMTP (Local Mail Transport Protocol, protocolo de transporte de e-mails locais) (veja a RFC2033). O padrão é o programa **mail.local** da distribuição **sendmail**. Ele é normalmente instalado no diretório **/usr/libexec**, porém podemos especificar o diretório de instalação através da opção `confEBINDIR`. A instalação-padrão do Red Hat não inclui **mail.local**. O SuSE o coloca em **/usr/lib/sendmail.d/bin** e o Debian o coloca em **/usr/lib/sm.bin**.

O recurso `smrsh` especifica o caminho para o programa a ser utilizado para entrega de e-mails a programas; o seu padrão é **/usr/libexec/smrsh**, o shell restrito de **sendmail**. O Red Hat instala o binário **smrsh** em **/usr/sbin**, o SuSE o coloca em **/usr/lib/sendmail.d/bin** e o Debian o coloca em **/usr/lib/sm.bin**. **smrsh** permite que o administrador de sistemas local controle quais comandos podem ser executados por e-mail.

Veja a seção 19.11 para uma discussão mais detalhada de **mail.local** e **smrsh**.

O recurso local_procmail

Você pode usr **procmail** de Stephen van den Berg como seu agente de entrega local habilitando o recurso `local_procmail`. Ele usa até três argumentos: o caminho para o binário **procmail**, o vetor de argumento para chamá-lo e flags para o agente de entrega. Os valores-padrão são razoáveis, porém o caminho-padrão (**/usr/local/bin/procmail**) conflita com a localização usual (**/usr/bin**) de **procmail** da maioria das distribuições.

procmail pode fazer coisas muito mais interessantes para o usuário do que o **/bin/mail** ou **mail.local** comum. Além de remeter e-mails para as caixas postais dos usuários, ele pode ordenar as mensagens em pastas, salvá-las em arquivos, executar programas e filtrar spams. O emprego do recurso `local_procmail` anula em grande parte as melhorias de segurança fornecidas por **smrsh** . Entretanto, caso não precise restringir os programas que seus usuários rodam (isto é, você confia em *todos* os seus usuários), então **procmail** pode ser muito prático. **procmail** não é distribuído com o **sendmail**; obtenha-o de www.procmail.org caso ele não seja instalado pelo seu fornecedor. Tanto o Red Hat, quanto o SuSE e o Debian incluem **procmail** e o colocam em **/usr/bin**.

Você pode usar outros programas para processamento de e-mail em conjunto com este recurso simplesmente mentindo para **sendmail** e dizendo que você está apenas mostrando a ele a cópia local de **procmail**:

```
FEATURE(`local_procmail', `/usr/local/bin/meu-agente-de-entrega')
```

Se usar **procmail**, experimente **/usr/bin/mailstat** para algumas estatísticas práticas sobre **procmail** (não se confunda com **/usr/sbin/mailstats**, que apresenta estatísticas de **sendmail**). Ele é instalado em todas as distribuições usadas como exemplo neste livro e pode ser utilizado para sintetizar arquivos de log de **procmail**. Não há nenhuma página de manual no Red Hat e há uma muito restrita no SuSE; o melhor que você tem a fazer nestas plataformas é ler o código-fonte (trata-se de um script de shell Bourne). Ou tente **mailstat -h** e experimente a partir dali. O Debian inclui uma página de manual adequada para **mailstat**.

As macros LOCAL_*

Se você realmente precisar sujar suas mãos e escrever algumas novas regras exóticas para lidar com situações locais especiais, use um conjunto de macros com o prefixo `LOCAL_`. A seção sobre spam, mais à frente, ainda neste capítulo, tem alguns exemplos desta construção de baixo nível.

Opções de configuração

As opções de arquivos de configuração e macros (os comandos `O` e `D` na linguagem de configuração bruta) podem ser configuradas por meio do comando `define` de **m4**. Uma lista completa das opções acessíveis como variáveis de **m4** e seus valores-padrão é dada no arquivo **cf/README**. Os valores-padrão são razoáveis para a maioria dos sites.

Snow666

Alguns exemplos:

```
define(`confTO_QUEUERETURN', `7d')
define(`confTO_QUEUEWARN', `4h')
define(`confPRIVACY_FLAGS', `noexpn')
```

A opção "queue return" determina por quanto tempo uma mensagem permanecerá na fila de e-mails caso ela não consiga ser entregue. A opção "queue warn" determina por quanto tempo uma mensagem permanecerá antes de o remetente ser notificado de que pode haver problemas com a entrega. As duas primeiras linhas configuram essas opções para, respectivamente, 7 dias e 4 horas.

Consulte a seção 19.11 para obter maiores informações sobre opções de privacidade.

A linha seguinte configura os flags de privacidade para desabilitar o comando SMTP EXPN (expandir endereço). A opção `confPRIVACY_FLAGS` usa uma lista de valores separados por vírgulas. Algumas versões de **m4** requerem dois conjuntos de aspas para proteger as vírgulas num campo com várias entradas, porém o GNU **m4** que vem com o Linux é mais inteligente e não requer tais aspas extras:

```
define(`confPRIVACY_FLAGS', ``noexpn, novrfy'')
```

Os valores-padrão para a maioria das opções são bem razoáveis para um site típico que não seja muito paranóico em relação a questões de segurança ou que não seja muito preocupado com o desempenho. Em particular, os padrões o protegem contra spams desativando a retransmissão, exigindo que endereços sejam totalmente qualificados e exigindo que os endereços tenham endereços IP. Se a sua máquina centralizadora de e-mail for muito ocupada e tiver que atender a um grande número de listas de discussão, talvez seja preciso ajustar alguns dos valores relativos a desempenho.

A Tabela 19.13 lista algumas opções que talvez você precise para ajustar (em torno de 15% de cerca de 175 opções de configuração), juntamente com seus valores-padrão. Para poupar espaço, os nomes das opções são mostrados sem seus prefixos `conf`; por exemplo, a opção `FALLBACK_MX` na verdade se chama `confFALLBACK_MX`. Dividimos a tabela em subseções que identificam o tipo de problema que a variável atende: genérico, recursos, desempenho, segurança e combate a spams e miscelânea. Algumas opções se encaixam claramente em mais de uma categoria, porém as listamos somente uma vez.

Tabela 19.13 Opções básicas de configuração.

	Nome da opção	Descrição e (valor-padrão)
Genérico	COPY_ERRORS_TO	Endereços para Cc em mensagens de erro (nenhum)
	DOUBLE_BOUNCE_ADDRESS	Captura uma grande quantidade de spams, alguns sites usam **/dev/null**, mas isto pode ocultar problemas sérios (postmaster)
Recursos	MIN_FREE_BLOCKS	Espaço de sistema de arquivos mínimo para aceitar e-mails (100)
	MAX_MESSAGE_SIZE	Tamanho máximo em bytes de uma única mensagem (infinito)
	TO_*lots_of_stuff*	Timeouts para todos os tipos de coisas (vários)
	TO_IDENT	Timeout para consultas ident para verificação da identidade do remetente; se for 0, as verificações ident não serão feitas (5s)
	MAX_DAEMON_CHILDREN	Número máximo de processos-filho[a] (nenhum limite)
Desempenho	MCI_CACHE_SIZE	Número de conexões TCP para fora abertas e colocadas em cache (2)
	MCI_CACHE_TIMEOUT	Tempo para manter abertas conexões em cache (5m)
	HOST_STATUS_DIRECTORY	Veja a página 460 para uma descrição (nenhum valor-padrão)
	FALLBACK_MX	Veja a página 460 para uma descrição (nenhum valor-padrão)
	FAST_SPLIT	Suprime pesquisas MX enquanto os receptores são ordenados e distribuídos entre filas; Veja a página 459 (1 = verdadeiro)
	QUEUE_LA	Carga média na qual os e-mails devem ser colocados em fila em vez de serem enviados imediatamente (8 * Num.CPUs)
	REFUSE_LA	Carga média na qual se deve recusar e-mails (12 * Num.CPUs)
	DELAY_LA	Carga média para reduzir a velocidade de entregas (0 = nenhum limite)
	MIN_QUEUE_AGE	Tempo mínimo que as tarefas têm de permanecer em fila; faz com que uma máquina muito ativa manipule melhor a fila (0)

www.snow666.blogspot.com
Snow666

Tabela 19.13 Opções básicas de configuração. (*continuação*)

	Nome da opção	Descrição e (valor-padrão)
Segurança/Spam	TRUSTED_USERS	Para proprietários de software de listas de discussão; permite forjar a linha From e reconstruir o banco de dados de nomes alternativos (root, daemon, uucp)
	PRIVACY_FLAGS	Limita as informações dadas por SMTP (avisos automáticos)
	INPUT_MAIL_FILTERS	Lista de filtros para e-mail que chegam (vazia)
	MAX_MIME_HEADER_LENGTH	Tamanho máximo dos cabeçalhos MIME (nenhum limite)[b]
	CONNECTION_RATE_THROTTLE	Diminui os ataques DOS limitando a velocidade na qual conexões de e-mails são aceitas (nenhum limite)
	MAX_RCPTS_PER_MESSAGE	Diminui a entrega de spams; protela receptores extras e envia uma mensagem de erro temporária (infinito)
	DONT_BLAME_SENDMAIL	Prevalece sobre a verificação de arquivos e segurança de **sendmail**; não altere irresponsavelmente! (segurança)
	AUTH_MECHANISMS SMTP	Mecanismos de autorização para o Cyrus SASL ([c])
Miscelânea	LDAP_DEFAULT_SPEC	Especificação de mapas para bancos de dados LDAP, inclusive o host e a porta que o servidor está executando (indefinido)

a. Mais especificamente, o número máximo de processos-filho que podem ser executados de uma só vez. Quando o limite for atingido, **sendmail** recusa conexões. Esta opção pode impedir (ou criar) ataques DOS (Denial Of Service).
b. Esta opção pode impedir estouros de buffer de agente de usuário. "256/128" é um valor indicado para uso — ele significa 256 bytes por cabeçalho e 128 bytes por parâmetro para aquele cabeçalho.
c. O valor-padrão é "EXTERNAL GSSAPI KERBEROS_V4 DIGEST-MD5 CRAM-MD5"; não acrescente PLAIN LOGIN a menos que você queira reduzir a segurança.

19.9 Exemplos de Arquivos de Configuração

Na verdade não terminamos de ver todas as opções de configuração (estão por vir uma seção sobre spam, uma sobre segurança e outra sobre desempenho), mas é chegado o momento de dar uma olhada em exemplos de arquivos de configuração. Essa ordem pode dar a impressão de colocarmos o carro na frente dos bois, porém dos arranjos que experimentamos, esse parece ser o melhor. O recurso de configuração essencial que não cobrimos ainda é o banco de dados de acesso, usado basicamente para filtro e controle de spams. Ele é visto a partir da seção 19.10.

Ao documentarmos os arquivos de configuração em uso em vários sites, invariavelmente nos deparamos com coisas antigas que estão erradas ou que não são mais necessárias. Limpamos essas inconsistências, erros de grafia etc., antes de usar os arquivos de configuração como exemplos, para que eles não sejam mostrados aqui exatamente como aparecem em seus servidores originais. Não obstante, eles reflitam configurações do mundo prático.

Um computador doméstico de um estudante de computação

Nosso primeiro exemplo é um estudante, Rob Braun, que tem um computador com Linux (gw.synack.net) em casa e faz hospedagem virtual para os domínios de alguns amigos: xinetd.org, teich.net e cubecast.com.

Ele também hospeda o seu próprio domínio, synack.net. Rob associa todos os e-mails que chegam para a pessoa correta via LDAP. Ele usa o recurso `virtusertable` para manipular os mapeamentos de hospedagem virtual e o recurso `genericstable` para manipular e-mails de saída. Dos diversos tipos de tabelas que afetam e-mails de saída, `genericstable` é o único capaz de reescrever o nome de usuário, bem como o host de destino.

O arquivo `genericstable` de Rob (que ele na verdade chama **`outmap`**) contém:

```
bbraun      rob@synack.net
stabilej    jon@synack.net
teich       oren@teich.net
```

Rob usa o Mail Abuse Prevention Service (`dnsbl`) para controle de spams. Ele também utiliza recursos de mascaramento para estampar qualquer e-mail de saída que já não esteja reescrito por `genericstable` como proveniente de usuario@synack.net em vez de usuario@gw.synack.net.

Snow666

Eis o arquivo **gw.mc** completo:

```
divert(0)
VERSIONID(`@(#)synack.net.mc 8.7 (Berkeley)5/19/1998')
OSTYPE(linux)
DOMAIN(generic)
FEATURE(dnsbl)
FEATURE(virtusertable, `hash /etc/mail/inmap')
FEATURE(genericstable, `hash /etc/mail/outmap')
GENERICS_DOMAIN_FILE(`/etc/mail/local-host-names')
MASQUERADE_AS(synack.net)
FEATURE(`masquerade_envelope')
FEATURE(`ldap_routing')
LDAPROUTE_DOMAIN(`synack.net')
define(`confLDAP_DEFAULT_SPEC', `-h gw.synack.net -b dc=synack,dc=net')
MAILER(local)
MAILER(smtp)
```

O arquivo **/etc/mail/local-host-names** contém os hosts e os domínios para os quais este host aceita e-mails. O recurso `use_cw_file` que chamaria o arquivo é oculto no arquivo de domínio genérico; veja a seção 19.9 para uma listagem de seu conteúdo. A retransmissão é desativada como padrão, pois o arquivo **/etc/mail/relay-domains** normalmente está vazio. Aqui, este arquivo contém os domínios que são hospedados virtualmente por gw.synack.net. O banco de dados LDAP é configurado com um arquivo **ldap.conf** que configura o host servidor, a porta e o nome root distinto:

```
BASE dc=synack, dc=net
HOST gw.synack.net
PORT 389
```

A seguir é construído o banco de dados LDAP por meio de um arquivo de texto com entradas como:

```
dn: uid=rob, dc=synack, dc=net
objectClass: inetLocalMailRecipient
mailLocalAddress: rob@synack.net
mailRoutingAddress: bbraun@synack.net
uid:rob

dn: uid=webmaster, dc=synack, dc=net
objectClass: inetLocalMailRecipient
mailLocalAddress: webmaster@synack.net
mailRoutingAddress: bbraun@synack.net
uid:webmaster

dn: uid=teich, dc=synack, dc=net
objectClass: inetLocalMailRecipient
mailLocalAddress: teich@synack.net
mailRoutingAddress: oren@teich.net
uid:teich

dn: uid=xinetd, dc=synack, dc=net
objectClass: inetLocalMailRecipient
mailLocalAddress: xinetd@synack.net
mailRoutingAddress: xinetd
uid:xinetd
```

As três primeiras entradas associam os nomes de login rob, webmaster e oren a seus nomes alternativos. A quarta é uma lista de discussão que é associada a um nome alternativo local e é manipulada lá por Majordomo por meio destas entradas no arquivo **/etc/mail/aliases**:

```
xinetd: "|/usr/local/majordomo/wrapper resend -l test xinetd-list"
xinetd-list: :include:/usr/local/majordomo/lists/xinetd
xinetd-owner: bbraun
owner-xinetd: bbraun
xinetd-request: bbraun
xinetd-approval: bbraun
```

Snow666

Nós mostramos vários arquivos de suporte e modelos de conteúdo para esse exemplo. Nossos próximos exemplos se mantêm mais próximos do objetivo de ilustrar como funciona a configuração de **sendmail**, e não a de mostrar arquivos periféricos.

Tenha em mente que em qualquer exemplo de **sendmail**, os registros MX do DNS desempenham um papel crucial e precisam concordar com as suposições feitas pela sua configuração.

Uma empresa pequena mas muito bem informada em termos de sendmail

Como próximo exemplo, veremos os arquivos e configuração para uma empresa pequena, mas muito inteligente em termos de **sendmail**, a Sendmail, Inc. O seu projeto de e-mails inclui uma máquina central de e-mail tanto para e-mails de entrada e de saída. Todos os e-mails são aceitos e direcionados imediatamente a um conjunto de servidores IMAP internos que verificam a existência de vírus em cada mensagem antes de liberá-la para a caixa postal de um usuário. A máquina centralizadora de e-mail também verifica a eventual existência de vírus para cada mensagem a ser enviada, de modo que Sendmail, Inc. jamais será responsável por disseminar vírus por e-mail. Analisamos primeiramente a configuração dos clientes, depois inspecionamos as máquinas mestras mais complexas.

Em todos os exemplos, modificamos ligeiramente os originais, eliminando os textos referentes a direito autoral, eventualmente acrescentando alguns comentários e excluindo a diretiva `dnl` do **m4** nos finais das linhas. Se você usar qualquer um de nossos exemplos como modelo para o seu arquivo **.mc**, certifique-se de suprimir os comentários no final das linhas.

Máquinas-cliente em sendmail.com

O arquivo **smi-client.mc** para máquinas-cliente é bem simples. Ele usa a máquina mestra smtp.sendmail.com que é, na verdade, apenas um outro nome para foon.sendmail.com. Usar um registro MX (ou um CNAME[35]) para apontar para o servidor de e-mails é uma boa idéia; é fácil de alterar quando você quiser mudar sua máquina mestra para e-mails.

Observe que a data neste arquivo é outubro de 1998. **sendmail** foi atualizado várias vezes desde então, porém o arquivo de configuração não precisou ser modificado.

```
divert(-1)
##### Este arquivo contém definições para o arquivo .mc
##### de uma máquina-cliente da Sendmail, Inc.
divert(0)
VERSIONID(`@(#)smi-client.mc 1.0 (Sendmail) 10/14/98')
OSTYPE(`bsd4.4')
FEATURE(`nocanonify')
undefine(`ALIAS_FILE')
define(`MAIL_HUB', `smtp.sendmail.com')
define(`SMART_HOST', `smtp.sendmail.com')
define(`confFORWARD_PATH', `')
MAILER(`local')
MAILER(`smtp')
```

As linhas `MAIL_HUB` e `SMART_HOST` direcionam e-mails que chegam e que saem para o host smtp.sendmail.com. Registros MX em DNS devem cooperar e listar aquele host de maior prioridade (o número mais baixo no registro MX) em vez das máquinas-cliente individuais. O caminho para os arquivos **.forward** são configurados em null e o arquivo de nome alternativo também é configurado em null; toda a expansão de nomes alternativos ocorre na máquina mestra. O recurso `nocanonify` é especificado aqui para poupar tempo, já que as pesquisas DNS são feitas de qualquer maneira no mestre.

Máquina mestra em sendmail.com

A máquina mestra em sendmail.com pode ser uma das instalações de **sendmail** existentes mais atacadas. Ela tem de ser segura contra todos os tipos de ataques de entrega de e-mail que as pessoas possam engendrar e tem de proteger as máquinas por trás dela. Eis seu arquivo de configuração:

```
divert(-1)
# Criado com o produto comercial Sendmail Switch da sendmail.com
divert(0)
```

35. Um registro MX é, na verdade, mais eficiente do que um CNAME; os CNAMEs requerem uma segunda pesquisa no nome verdadeiro para obter o endereço IP.

www.snow666.blogspot.com
Snow666


```
ifdef(`COMMERCIAL_CONFIG', `INPUT_MAIL_FILTER(`mime-filter', `S=local:
   /var/run/mime-filter/mime-filter.sock')')
LDAPROUTE_DOMAIN(`sendmail.com sendmail.net sendmail.org')
MASQUERADE_AS(`sendmail.com')
MASQUERADE_DOMAIN(`sendmail.com')
RELAY_DOMAIN(`sendmail.com sendmail.net sendmail.org')
define(`MAIL_HUB', `internal-hub.sendmail.com')
define(`QUEUE_DIR', `/var/spool/mqueue/q*')
define(`SMART_HOST', `virus-scan.sendmail.com')
ifdef(`COMMERCIAL_CONFIG', `define(`confCACERT', `/local/certs/cacert.pem')')
ifdef(`COMMERCIAL_CONFIG', `define(`confCACERT_PATH', `/local/certs/trustedcerts')')
define(`confCHECK_ALIASES', `True')
ifdef(`COMMERCIAL_CONFIG', `define(`confCLIENT_CERT', `/local/certs/cert.pem')')
ifdef(`COMMERCIAL_CONFIG', `define(`confCLIENT_KEY', `/local/certs/key.pem')')
define(`confEIGHT_BIT_HANDLING', `mimify')
define(`confLDAP_DEFAULT_SPEC', ` -h "ldap.sendmail.com ldap2.sendmail.com"
   -b "dc=sendmail,dc=com" -p 1389')
define(`confREFUSE_LA', `99')
define(`confRUN_AS_USER', `mailnull')
ifdef(`COMMERCIAL_CONFIG', `define(`confSERVER_CERT', `/local/certs/cert.pem')')
ifdef(`COMMERCIAL_CONFIG', `define(`confSERVER_KEY', `/local/certs/key.pem')')
define(`confTO_IDENT', `0s')
define(`confTO_QUEUEWARN', `2d')
ifdef(`confPOP_TO', `', `define(`confPOP_TO', `900')')
FEATURE(`accept_unqualified_senders')
FEATURE(`accept_unresolvable_domains')
FEATURE(`allmasquerade')
FEATURE(`always_add_domain')
FEATURE(`domaintable')
FEATURE(`ldap_routing', `ldap -1 -v mailHost -k ldap -1 -v mailhost -k
   (&(objectclass=mailRecipient)(|(mail=%0)(|(mailAlternateAddress=%0))))',
   `ldap -1 -v mail -k
   (&(objectclass=mailRecipient)(|(mailalternateaddress=%0)))', `passthru')
FEATURE(`mailertable')
FEATURE(`masquerade_entire_domain')
FEATURE(`masquerade_envelope')
FEATURE(`relay_entire_domain')
FEATURE(`use_cw_file')
MAILER(`local')
MAILER(`smtp')

LOCAL_RULESETS
SLocal_check_rcpt
R$*    $: $&{verify}
ROK    $# OK
```

A tarefa da máquina mestra é a de direcionar os e-mails que chegam para o servidor interno adequado e a de servir como o host de retransmissão inteligente para e-mails que saem.

Devido às duas linhas a seguir,

```
FEATURE(`accept_unqualified_senders')
FEATURE(`accept_unresolvable_domains')
```

todos os e-mails que chegam são aceitos, mesmo e-mails de remetentes não qualificados e de domínios não resolvíveis. Dessa forma, clientes em potencial que possuem **sendmail** ou DNS mal configurado ainda podem passar. Essas regras desfazem os padrões que capturam uma grande quantidade de spams com cabeçalhos forjados. Ident é desativada (timeout configurado em 0) para acelerar a entrega de e-mail que chega.

Esta máquina mestra de e-mails verifica primeiramente as mensagens que chegam para determinados tipos de anexos MIME freqüentemente usados por vírus (instrução `INPUT_MAIL_FILTER`). O **mime-filter** chamado lá contém linhas como:

```
*:anniv.doc:* error:Your email was not accepted by Sendmail, it appears to be
   infected with the Melissa-X virus.
```

Snow666


```
*:.vbs:*     error:For security and virus protection reasons, Sendmail does
   not accept messages with VBS files attached. Please retransmit your
   message without the VBS file.
```

Anexos MIME do tipo **.vba**, **.dot**, **.exe**, **.com**, **.reg** e assim por diante são rejeitados, mas uma varredura completa contra vírus não é feita aqui, pois ela diminuiria a velocidade de processamento de e-mails que chegam. O mestre usa LDAP (com um esquema específico ao site) para pesquisar o receptor de cada mensagem e direcioná-la para o servidor IMAP/POP interno correto. Se o receptor não for encontrado no banco de dados LDAP, o e-mail será enviado para uma máquina mestra interna (a instrução `MAIL_HUB`) para processamento adicional. Tanto os servidores IMAP/POP quanto a máquina mestra interna realizam varredura contra vírus antes de encaminharem uma mensagem para a caixa postal de um usuário.

E-mails que saem também são direcionados por meio desta máquina mestra por instruções `SMART_HOST` nas máquinas-cliente. Para enviar uma mensagem via servidores de e-mail sendmail.com, os hosts fora do domínio sendmail.com têm de apresentar um certificado assinado pela autoridade de certificação de sendmail.com. Funcionários visitando o site de um cliente podem retransmitir e-mails para um terceiro por meio de sendmail.com com este mecanismo, porém outros não poderão fazê-lo. Essa convenção autentica cada usuário e impede que e-mail forjado transite por sendmail.com.

Após aceitar e-mails destinados à Internet, a máquina mestra os passa para `SMART_HOST` para verificação da existência de vírus. A máquina mestra de e-mails não é tão ocupada para não poder ela própria fazer a varredura contra vírus, porém, se a varredura fosse feita aí, usuários enviando e-mails teriam de esperar a varredura ser completada antes de suas mensagens realmente serem enviadas. Jogá-las em fila na máquina de varredura de vírus deixa os usuários felizes — suas mensagens parecem partir instantaneamente.

As regras `LOCAL_CONFIG` no final do arquivo de configuração são o local onde a verificação de cabeçalhos contra vários vírus e spammers conhecidos são normalmente colocados. Podem ser encontrados bons exemplos no arquivo **knecht.mc** da distribuição de **sendmail**. Incluímos um exemplo a seguir.

*Consulte a seção 19.10 para obter maiores detalhes sobre o **libmilter**.*

Durante o verão de 2001, o arrasador worm SirCam estava circulando ferozmente. O trecho a seguir extraído do arquivo **knecht.mc** na distribuição **sendmail** o captura. O SirCam é uma das primeiras mensagens sórdidas a ter cabeçalhos aleatórios. As ferramentas usuais para capturá-lo teriam falhado, exceto pelo fato de seus autores terem cometido um erro que diferenciava uma mensagem SirCam de uma verdadeira mensagem Outlook Express. O conteúdo da mensagem é bem comum (ele solicita sua atenção ao anexo contido na mensagem) e seria, portanto, um candidato para os recursos de filtragem do novo **libmilter** da versão 8.12. Sem garantias de responsabilidade do produto no mundo do software, parece que a única solução para todos estes vírus e worms Microsoft é jogar fora o Windows e instalar o Linux em tudo quanto é canto!

```
LOCAL_RULESETS

KSirCamWormMarker regex -f -aSUSPECT multipart/mixed;boundary=----
   .+_Outlook_Express_message_boundary
HContent-Type: $>CheckContentType

SCheckContentType
R$+                 $: $(SirCamWormMarker $1 $)
RSUSPECT            $#error $: "553 Possible virus, see http:
   //www.symantec.com/avcenter/venc/data/w32.sircam.worm@mm.html"

HContent-Disposition:$>CheckContentDisposition

SCheckContentDisposition
R$-                 $@ OK
R$- ; $+            $@ OK
R$*                 $#error $: "553 Illegal Content-Disposition"
```

Clientes em sendmail.com não possuem nenhum controle contra spams em seus arquivos de configuração, pois todos os e-mails que chegam ao site são provenientes do centralizador de e-mails externo e de um centralizador interno, e spam é filtrado lá. Alguns dos recursos e outras construções nesse exemplo não são vistos em nossa seção sobre configuração, mas você poderá localizar a documentação sobre eles no arquivo **cf/README**.

Snow666

Um outro exemplo mestre/cliente

XOR Inc. é uma empresa de médio porte com uma única máquina mestra para e-mails. Embora o projeto para e-mails como um todo da XOR seja similar àquele da sendmail.com, ele é implementado com primitivas de configuração ligeiramente diferentes.

Eis a configuração do cliente:

```
divert(-1)
##### xor-client.mc, todos os clientes retransmitem para xor.com
divert(0)
VERSIONID(`@(#)tcpproto.mc8.5 (Berkeley) 3/23/96')
OSTYPE(`bsdi')
define(`confPRIVACY_FLAGS', `noexpn')
FEATURE(`nullclient', `xor.com')
```

Ele realmente é mínimo, porém excelente para um projeto de e-mails no qual o servidor executa todas aquelas coisas extravagantes e os clientes são simplesmente capazes de enviar e-mail. Até mesmo e-mails para usuários locais são encaminhados para xor.com, a máquina especificada no recurso `nullclient`. Não é especificado nenhum agente de entrega; **sendmail** não é executado no modo daemon.

A seguir temos a configuração mestra que vai com a configuração de cliente. XOR realiza um bocado de hospedagem Web e aceita e gerencia e-mails para domínios virtuais. Nosso primeiro exemplo, o da máquina de um estudante, gerenciava três domínios virtuais com LDAP e o recurso `genericstable`. XOR gerencia cerca de 1.000 domínios virtuais com o recurso `virtusertable`. Ele usa o recurso `genericstable` para associar nomes de login a *primeiro.ultimo* para e-mails que saem. Ele implementa nomes alternativos com o arquivo **aliases** padrão, com 3.000 linhas de comprimento e contém várias listas de discussão, algumas com milhares de receptores (e uma com mais de 100.000). Tudo isso se encontra num velho computador sun4m com SunOS ligeiramente fatigado que não é substituído por um computador Linux novo e sofisticado devido à inércia e um pouco da filosofia "em time que está ganhando não se mexe".

Observe que as instruções `divert` e os comentários que normalmente estão presentes no início de um arquivo **.mc** estão ausentes. Eles são necessários somente se você usar comentários tipo shell (#) no início de seu arquivo de configuração.

Este site está rodando **sendmail** 8.9.3 e usa algumas das construções antigas (anteriores à versão 8.10) para linhas de configuração. Sua carga de e-mails fez com que muitos dos parâmetros referentes a desempenho fossem configurados com valores bem maiores do que seus padrões tradicionais.

```
VERSIONID(`@(#)xor.mc3.0 (trent) 10/7/01')
OSTYPE(`sunos4.1')

define(`confPRIVACY_FLAGS', `noexpn,novrfy')
define(`confMESSAGE_TIMEOUT',`5d/72h')
define(`LOCAL_MAILER_PATH', `/usr/bin/mail.local')

dnl ##### aumenta valores para fins de desempenho e carga pesada
define(`confMCI_CACHE_SIZE', `16')
define(`confMCI_CACHE_TIMEOUT', `10m')
define(`confCHECK_ALIASES', `False') ### valor-padrão, não necessário
define(`confDOMAIN_NAME', `xor.com')
define(`confMAX_MESSAGE_SIZE', `5000000')
define(`confDAEMON_OPTIONS',`Port=NNN') ### NNN tem que estar em /etc/services
define(`confQUEUE_LA', 25)
define(`confREFUSE_LA', 30)

FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(virtusertable)
GENERICS_DOMAIN(`xor.com')
FEATURE(genericstable)
FEATURE(`masquerade_envelope')
FEATURE(`redirect')
FEATURE(`access_db', `hash -o /etc/mail/access')

MAILER(local)
```

Snow666


```
MAILER(smtp)

LOCAL_RULESETS
# Rotina para verificação da existência de vírus Melissa de Kludgey que verifica cabeçalhos em termos
de padrão.
# Nota instrutiva: o formato para a regra é
#               RExatamente aquilo que você está querendo colocar entre aspas
# Nenhuma aspas, nenhuma tabulação, absolutamente nada entre parênteses, depois  então a
# coisa exata, a seguir uma tabulação e o $#erro que seleciona o agente de entrega a ser usado.
# Note, que $* concorda com qualquer coisa, portanto, é útil para caracteres-curinga; sendmail deve
# varrer todas as mensagens assim haverá um pequeno impacto sobre o desempenho.

Hsubject:          $>local_check_header
D{MelissaMessage}"553 Your message may contain the Melissa virus. Please
   e-mail postmaster@$j if you have questions."
D{LoveVirus}"553 Your message probably contains the LOVELETTER virus -
   Please email postmaster@$j if you have questions "
D{virus}"553 We think that your message contains a VIRUS so we will reject it"

Slocal_check_header
RPeace between America and Islam! $*  $#error $: ${virus}
RPEACE BETWEEN AMERICA AND ISLAM! $*  $#error $: ${virus}
RPeace between America and Islam! $*  $#error $: ${virus}
RPeace between America and Islam! $*  $#error $: ${virus}
RPeace between America and Islam! $*  $#error $: ${virus}
RWar vote$*                           $#error $: ${virus}
Rwar vote$*                           $#error $: ${virus}
Rimportant Message From $*            $#error $: ${MelissaMessage}
RRe: Important Message From $*        $#error $: ${MelissaMessage}
RILOVEYOU                             $#error $: ${LoveVirus}
RI LOVE YOU                           $#error $: ${LoveVirus}
Rfwd: Joke                            $#error $: ${virus}
Rsusitikim$*                          $#error $: ${virus}
R$*Jokes text                         $#error $: ${virus}
R$*Jokes                              $#error $: ${virus}
R$*Funny text                         $#error $: ${virus}

dnl ##### eliminadas algumas centenas de linhas localizando vírus e spam

R$*The declaration of independence$*              $#error $: ${spam}
R$*Affordable Benefits$*                          $#error $: ${spam}
R$*Do your own Criminal Background Checks$*       $#error $: ${spam}
R$*Are YOU Ready For Wealth & Freedom ????$*      $#error $: ${spam}
```

As regras contra vírus e spams no final do arquivo **.mc** rejeitam mensagens específicas tomando como base suas linhas de assunto. Embora estas funcionem bem, as variações de letras maiúsculas e minúsculas não são necessárias; conjuntos de regras não diferenciam letras maiúsculas de minúsculas. Algumas das linhas de assunto alvo (por exemplo, `$*Jokes`) poderiam possivelmente ser concordantes com e-mails legítimos.

Essas regras afetam um pouco o desempenho de **sendmail** e, na velocidade em que o spam chega, poderia acabar deixando o arquivo de configuração enorme. Caso o seu site seja muito ocupado, provavelmente seja melhor fazer a filtragem de spams em uma outra máquina em vez de no seu gateway de e-mails principal. Para eliminar todos os e-mails de sites específicos, use o banco de dados de acesso e a palavra-chave `REJECT`.

O arquivo de acesso em xor.com contém mais de 800 domínios dos quais rejeitar e-mails, a maioria deles sites de pornografia e de spam a julgar pelos nomes de seus domínios. Eis um pequeno trecho:

```
adult-megasite.com       REJECT
adult-zone.com           REJECT
adult10000.com           REJECT
adultage.com             REJECT
. . .
vivarebates.com          REJECT
vividnet.com             REJECT
```

Snow666


```
vjs.org                 REJECT
vmadmin.com             REJECT
vnuemedia.com           REJECT
. . .
```

Como pode ser observado pela diversidade desses exemplos, não há só uma única maneira para definir um arquivo de configuração. **sendmail** contém muitas construções para direcionamento de e-mails e mudanças de cabeçalhos. Até certo ponto, aquelas escolhidas por você dependem de preferência pessoal ou daquilo que a pessoa da qual você copiou o arquivo fez.

Agora que já vimos alguns exemplos práticos de configurações, vamos dar uma olhada nas configurações-padrão que acompanham cada um das distribuições Linux usadas como referência neste livro.

A configuração de sendmail no Red Hat

O Red Hat vem com **sendmail** instalado. Como padrão, ele emprega uma configuração construída por meio de **/etc/mail/sendmail.mc** e utiliza as demais partes da árvore de configuração do diretório **/usr/lib/sendmail-cf**, que parece ser o diretório **cf** da distribuição **sendmail**.

O arquivo **.mc** do Red Hat usa nomes e localizações antigas para arquivos de **sendmail** em vez dos padrões atuais em **/etc/mail**. A configuração inclui diversos dos bancos de dados que controlam o direcionamento de e-mails. Entretanto, esses arquivos precisam ser preenchidos com dados locais; **mailertable**, **virtusertable** e **access** existem, mas estão vazios. Um arquivo, **userdb**, é configurado mas está ausente.

Nós eliminamos alguns dos comentários (as linhas começando com `dnl`) para poupar espaço e também adicionamos comentários (começando com `###`) em linhas incomuns. Você teria de eliminar nossos comentários antes de usar o exemplo como um arquivo **.mc** real. Eis o arquivo **.mc** do Red Hat conforme distribuído no início de 2002:

```
divert(-1)
include(`/usr/lib/sendmail-cf/m4/cf.m4')            ### maneira antiga de construir o arquivo cf
VERSIONID(`linux setup for Red Hat Linux')
OSTYPE(`linux')
define(`confDEF_USER_ID',``8:12'')                  ### uid/gid para 'mail' de usuário
undefine(`UUCP_RELAY')                              ### desnecessário, padrão
undefine(`BITNET_RELAY')                            ### desnecessário, padrão
define(`confAUTO_REBUILD')                          ### em desuso, veja abaixo
define(`confTO_CONNECT', `1m')
define(`confTRY_NULL_MX_LIST',true)                 ### ruim e perigoso
define(`confDONT_PROBE_INTERFACES',true)            ### estranho, veja abaixo
define(`PROCMAIL_MAILER_PATH',`/usr/bin/procmail')  ### duplicado, em ostype
define('ALIAS_FILE','/etc/aliases')
dnl define(`STATUS_FILE', `/var/log/statistics')    ### colocado como comentário, RH
   7.1 tinha um erro de nome de arquivo aqui
define(`UUCP_MAILER_MAX', `2000000')                ### mailer uucp não é usado
define(`confUSERDB_SPEC', `/etc/mail/userdb.db')    ### em desuso, veja abaixo
define(`confPRIVACY_FLAGS', `authwarnings,novrfy,noexpn,restrictqrun')
define(`confAUTH_OPTIONS', `A')
FEATURE(`no_default_msa',`dnl')                     ### mata o MSA padrão mas
   não há substituto especificado com DAEMON_OPTIONS
FEATURE(`smrsh',`/usr/sbin/smrsh')
FEATURE(`mailertable',`hash -o /etc/mail/mailertable.db')
FEATURE(`virtusertable',`hash -o /etc/mail/virtusertable.db')
FEATURE(redirect)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(use_ct_file)
FEATURE(local_procmail,`',`procmail -t -Y -a $h -d $u')
FEATURE(`access_db',`hash -o /etc/mail/access.db')
FEATURE(`blacklist_recipients')
EXPOSED_USER(`root')
dnl Isto muda o sendmail para escutar somente no dispositivo de loopback  127.0.0.1
dnl e não em nenhum outro dispositivo de rede. Coloque como comentário se você quer
dnl aceitar e-mail através da rede
DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')
dnl NOTA: associar ambos os daemons de IPv4 e de IPv6 na mesma porta requer
```

www.snow666.blogspot.com
Snow666


```
dnl   uma modificação ao kernel
dnl DAEMON_OPTIONS(`port=smtp,Addr=::1, Name=MTA-v6, Family=inet6')
dnl Nós recomendamos enfaticamente que se coloque isto como comentário se você quer proteger
dnl a si mesmo contra spam. Contudo, laptop e usuários em computadores que não
dnl tenham um DNS 24x7 precisam disto.
FEATURE(`accept_unresolvable_domains')          ### deve ser eliminado, veja abaixo
dnl FEATURE(`relay_based_on_MX')                ### depositada muita confiança, veja a seção sobre spams
MAILER(smtp)
MAILER(procmail)
Cwlocalhost.localdomain                         ### deve usar nomes de host locais
```

A opção `confAUTO_REBUILD` está caindo em desuso e será eliminada em versões futuras. Ela faz com que **sendmail** reconstrua seu banco de dados de nomes alternativos caso ele tenha sido alterado antes de a manipulação de qualquer e-mail.

A definição `confDONT_PROBE_INTERFACES` é estranha; ela faz com que **sendmail** não liste as interfaces da máquina local em sua classe `w` (**/etc/mail/local-host-names**). Os e-mails têm então de ser manipulados de forma especifica por uma entrada `mailertable`, ou ele irá ricochetear com um erro de configuração. Mesmo se `confDONT_PROBE_INTERFACES` estiver ativo, a interface loopback será sempre inclusa, de modo que e-mail local deve funcionar. (Entretanto, não é definido nenhum agente de entrega local e, portanto, talvez não funcione.)

O Red Hat 7.1 especificava **sendmail.st** como `STATUS_FILE`, porém criou o arquivo **statistics**, e nenhum dado estatístico era efetivamente coletado. O Red Hat 7.2 corrigiu o nome do arquivo, mas colocou a linha como comentário.

O recurso `userdb` está caindo em desuso e também não é utilizado nesta configuração, de modo que a linha `confUSERDB_SPEC` é estranha, talvez um resquício do passado. Tanto o recurso `accept_unresolvable_domains` quanto o recurso `relay_based_on_MX` devem ser eliminados, se possível; o primeiro cria abrigos para spams e o segundo permite que o spammer roube serviço.

Configuração de sendmail no SuSE

O SuSE vem com duas configurações, ambas em **/etc/mail**: **linux.mc** e **linux.nullclient.mc**. Uma é para emprego numa máquina mestra de e-mails e a outra para máquinas-cliente. A grande diferença nas versões do SuSE na forma como vêm de fábrica está no arquivo `OSTYPE`, **suse-linux.m4**, que contém de longe muita informação.

Os arquivos **.mc** e os arquivos de modelos de bancos de dados contêm toneladas de comentários para ajudá-lo a entender que recursos são incluídos e o formato para informações locais. O arquivo **linux.mc** mostrado a seguir foi reduzido de 220 linhas para 22 linhas após eliminarmos os comentários.

A lista de agentes de entrega inclui alguns que não fazem parte da distribuição normal de **sendmail**. Eles podem ser encontrados na versão fornecida em **/usr/share/sendmail/mailers**.

O arquivo **.mc** do SuSE viola a ordem esperada por **sendmail** nas instruções de configuração colocando as instruções `DOMAIN` (que normalmente se encontram próximas do início) após os agentes de entrega (que normalmente são os últimos). Se uma instrução no arquivo `DOMAIN` modificar um parâmetro mailer, esta modificação não entrará em efeito devido à ordem das instruções. É melhor seguir a ordem sugerida no arquivo **cf/README**. Embora os arquivos SuSE pareçam interessantes, organizados e simples, a maior parte da configuração está oculta no arquivo OSTYPE do **suse-linux**. Eis o arquivo **linux.mc**:

```
divert(-1)
# Copyright (c) 1997-1999,2000 SuSE GmbH Nuernberg, Germany.
# Author: Florian La Roche
#         Werner Fink <werner@suse.de>
# Após o  `divert(0)' todas as linhas começando com  `dnl' são
# comentários até o próximo caractere de nova linha.
# Colocando palavras entre pares `'- desabilita a expansão macro
### deve ser feito em ./Build, tem que estar após divert(0)
include(`/usr/share/sendmail/m4/cf.m4')
divert(0)
VERSIONID(`@(#)Setup for SuSE Linux         8.11.0-0.4 (SuSE Linux) 04/09/2000')
OSTYPE(`suse-linux')
FEATURE(`always_add_domain')
MAILER(`local')
MAILER(`smtp')
MAILER(`procmail')
MAILER(`uucp')
```

Snow666


```
MAILER(`bsmtp')
MAILER(`fido')
DOMAIN(`generic')
LOCAL_CONFIG          ### inútil, não faz nada
```

E eis o arquivo **linux.nullclient.mc**; você tem de substituir mailhub.domain.notused pelo nome verdadeiro de sua máquina centralizadora de e-mails. Eliminamos `dnl` nos finais das linhas para fazer com que o exemplo ficasse mais fácil de ser lido.

```
divert(-1)
# Copyright (c) 1999 SuSE GmbH Nuernberg, Germany.
# Author: Werner Fink <werner@suse.de>

# Este é um caso especial -- ele cria um arquivo de configuração reduzido
# contendo nada exceto suporte para encarminhar todo o e-mail para um
# centralizador via uma rede local por SMTP. O argumento é o
# nome deste centralizador

### deve ser feito em ./Build, tem que estar após divert(0)
include(`/usr/share/sendmail/m4/cf.m4')
divert(0)
VERSIONID(`@(#)Setup for SuSE Linux        8.11.0-0.2 (SuSE Linux) 28/03/2000')
OSTYPE(`suse-linux')
FEATURE(`nullclient', `mailhub.domain.notused')
FEATURE(`nocanonify')
LOCAL_CONFIG                ### inútil
```

Arquivos `OSTYPE` têm, tipicamente, cerca de 10 linhas e contêm o caminho para diversos programas cuja localização é dependente do sistema operacional. O SuSE Linux extrapolou; novamente inserimos comentários iniciando com `###` em linhas incomuns. A única linha `OSTYPE(`suse-linux')` faz com que o arquivo a seguir seja incluso nestas configurações simples:

```
divert(-1)
# Copyright (c) 1999,2000 SuSE GmbH Nuernberg, Germany.
# Author: Werner Fink <werner@suse.de>

divert(0)
VERSIONID(`@(#) suse-linux.m4 8.11.1-0.5 (SuSE Linux) 22/11/2000')
define(`confCF_VERSION', `SuSE Linux 8.11.1-0.5')

define(`confDEF_USER_ID', `daemon:daemon')        ### inseguro se algo mais usar uid/gid da daemon
define(`confCOPY_ERRORS_TO', `Postmaster')
define(`UUCP_MAILER_MAX', `2000000')
define(`confMAX_MIME_HEADER_LENGTH', `256/128')
define(`confMAX_HEADERS_LENGTH', `32768')         ### padrão
define(`confQUEUE_LA', `12')
define(`confREFUSE_LA', `18')
define(`MAX_DAEMON_CHILDREN', `30')
define(`confTO_ICONNECT', `30s')
dnl Muitos administradores de sistema desabilitaram IDENT
define(`confTO_IDENT', `0s')
dnl Deveríamos configurar  noreceipts também conhecido como desabilite DSN?
define(`confPRIVACY_FLAGS',`authwarnings,needmailhelo,novrfy,noexpn,noverb')
define(`confTRUSTED_USERS', `mdom wwwrun root uucp daemon')
define(`confNO_RCPT_ACTION', `add-to-undisclosed')
dnl Nota: RFC1891 diz isto, mas é freqüentemente mal utilizado
dnl define(`confRRT_IMPLIES_DSN', `True')
FEATURE(`always_add_domain')

define(`PROCMAIL_MAILER_PATH', `/usr/bin/procmail')
define(`PROCMAIL_MAILER_ARGS', `procmail -m $h $f $u')
FEATURE(`local_procmail')
define(`LOCAL_SHELL_FLAGS', `u09')
```

Snow666


```
define(`LOCAL_MAILER_ARGS', `procmail -a $h -d $u')
define(`LOCAL_MAILER_FLAGS', `SPfhn09')
define(`USENET_MAILER_PATH', `/usr/bin/inews')

define(`DATABASE_MAP_TYPE', `hash')                    ### padrão

define(`SMLIBDIR', `/usr/lib/sendmail.d')
define(`MAIL_SETTINGS_DIR', `/etc/mail/')              ### padrão
define(`confDEAD_LETTER_DROP', `/var/log/dead.letter')

syscmd(`test -f /etc/SuSE-release')
ifelse(sysval, 0,
`define(`STATUS_FILE', `/var/log/sendmail.st')'
`define(`QUEUE_DIR', `/var/spool/mqueue')'             ### padrão
`define(`STATUS_FILE', `/etc/mail/sendmail.st')'       ### nome antigo para o arquivo stats
`define(`QUEUE_DIR', `/var/mqueue')'
)
define(`confHOST_STATUS_DIRECTORY', `.hoststat')
define(`HELP_FILE', SMLIBDIR`/helpfile')
define(`ALIAS_FILE', `/etc/aliases')
define(`confHOSTS_FILE', `/etc/hosts')                    ### padrão
define(`confCT_FILE', `-o /etc/mail/sendmail.ct %[^\#]')    ### %[^\#] desnecessário
define(`confCW_FILE', `-o /etc/mail/sendmail.cw %[^\#]')    ### %[^\#] desnecessário
define(`confCR_FILE', `-o /etc/mail/relay-domains %[^\#]')  ### %[^\#] desnecessário
define(`confUSERDB_SPEC', `/etc/mail/userdb.db') ### userdb está caindo em desuso
define(`confSERVICE_SWITCH_FILE', `/etc/mail/service.switch')### padrão
define(`confEBINDIR', SMLIBDIR`/bin')
define(`confDONT_BLAME_SENDMAIL', `AssumeSafeChown,TrustStickyBit') ###
   perigoso
define(`confCONTROL_SOCKET_NAME', `/var/run/sendmail/control')
FEATURE(`no_default_msa') ### estranho, veja abaixo

FEATURE(`mailertable', `hash -o /etc/mail/mailertable.db')
FEATURE(`genericstable', `hash -o /etc/mail/genericstable.db')
FEATURE(`virtusertable', `hash -o /etc/mail/virtusertable.db')
FEATURE(`access_db', `hash -o /etc/mail/access.db')
dnl
LOCAL_CONFIG          ### inútil
```

Isso é uma verdadeira bagunça. A maior parte dela pertence ao arquivo **.mc** (ou a um arquivo de domínio das primitivas de configuração para o seu site) onde se pode vê-la facilmente, não oculta por trás de uma camada de indireção `OSTYPE`. Os arquivos `OSTYPE` devem ser reservados para parâmetros dependentes do sistema operacional como o caminho para um determinado programa. O arquivo **cf/README** na distribuição **sendmail** inclui uma lista de definições que deveriam ser inclusas no arquivo `OSTYPE` — quem sabe alguém poderia enviar ao SuSE uma cópia realçada.

Apesar desse ridículo arquivo `OSTYPE`, o SuSE tem fornecido um arquivo de configuração inicial bem razoável para se basear. Todas as tabelas referidas como tendo arquivos correspondentes em **/etc/mail** com comentários indicam o formato e as entradas típicas. Essa configuração precisa apenas de um pouco de reorganização num arquivo `OSTYPE` apropriado e quem sabe um arquivo `DOMAIN` do SuSE, com verdadeiro conteúdo — os recursos usados — permanecendo no arquivo **.mc** ao qual eles pertencem. O recurso `no_default_msa` é um tanto estranho; ele desabilita a porta 587, que todos os agentes de usuário mais novos usam para submeter mensagens ao sistemas de e-mail. Não fica claro porque o SuSE age dessa forma.

Note as definições condicionais (introduzidas pela palavra-chave `ifelse` do **m4**) para a localização do arquivo de informações estatísticas e o diretório para filas. O arquivo **/etc/SuSE-release** realmente existe, de modo que a diretiva `syscmd` de **m4** é bem-sucedida e retorna 0 em `sysval`, fazendo com que a construção `ifelse` de **m4** selecione a primeira alternativa em vez da segunda. O SuSE não inclui uma página de manual para **m4** em sua distribuição, de modo que você talvez tenha de ir a uma outra máquina para realmente descobrir os detalhes destas condicionais.

www.snow666.blogspot.com
Snow666

A configuração de sendmail no Debian

O arquivo **sendmail.mc** do Debian é gerado pelo script de instalação e, como padrão, inclui mascaramento e não muita coisa a mais. Ele usa `OSTYPE(debian)`, que puxa o arquivo **/usr/share/sendmail/sendmail.cf/ostype/debian.m4** mostrado a seguir:

```
divert(0)
VERSIONID(`@(#)debian.m4         8.9.3-21 (Debian) 20000309')
define(`confCF_VERSION', `Debian 8.9.3-21')

# mudanças feitas aqui *devem* ser refletidas em parsemc,updatedb,debian.m4

# caminhos
define(`_USE_ETC_MAIL_',              `True') # um pouco tarde para alguns itens ;-{
define(`DATABASE_MAP_TYPE',           `hash') ### padrão
define(`STATUS_FILE',                 `/var/lib/sendmail/sendmail.st')
define(`confHOST_STATUS_DIRECTORY',   `/var/lib/sendmail/host_status')
define(`HELP_FILE',                   `/usr/share/sendmail/sendmail.hf')
define(`confCR_FILE',                 `-o /etc/mail/relay-domains %[^\#]')
   ### %[^\#] desnecessário
define(`confCT_FILE',                 `-o /etc/mail/trusted-users %[^\#]')
define(`confCW_FILE',                 `-o /etc/mail/local-host-names %[^\#]')
   ### %[^\#] desnecessário
define(`confUSERDB_SPEC',             `/etc/mail/users')          ### em desuso
define(`ALIAS_FILE',                  `/etc/mail/aliases')        ### padrão
define(`confSERVICE_SWITCH_FILE',     `/etc/mail/service.switch') ### padrão
define(`confEBINDIR',                 `/usr/lib/sm.bin')

# flags
define(`confCON_EXPENSIVE', `True')
define(`confDIAL_DELAY', `45')
define(`confMAX_DAEMON_CHILDREN', `30')
define(`confNO_RCPT_ACTION', `add-to-undisclosed')
define(`confQUEUE_SORT_ORDER', `Host')
define(`confSAFE_FILE_ENV')
define(`confSAFE_QUEUE', `True')
define(`confTEMP_FILE_MODE', `600')
define(`confTO_ICONNECT', `30s')

define(`LOCAL_MAILER_PATH', `/usr/lib/sm.bin/sensible-mda')
define(`LOCAL_MAILER_FLAGS', `hnPu90')
define(`LOCAL_MAILER_ARGS', `sensible-mda $g $u $h ${client_addr}')
define(`LOCAL_SHELL_FLAGS', `u90')
define(`PROCMAIL_MAILER_PATH', `/usr/bin/procmail')
define(`USENET_MAILER_PATH', `/usr/bin/inews')
define(`confMAX_MIME_HEADER_LENGTH', `256/128')
define(`confMAX_HEADERS_LENGTH', `32768')

# Usuários Debian têm diretórios/arquivos com permissão de gravação para grupo por padrão...
define(`confDONT_BLAME_SENDMAIL',
   `ClassFileInUnsafeDirPath,ForwardFileInGroupWritableDirPath,GroupWritab
   leAliasFile,GroupWritableForwardFileSafe,GroupWritableIncludeFileSafe,Incl
   udeFileInGroupWritableDirPath,MapInUnsafeDirPath') ### acaba com a segurança
#
DOMAIN(generic)
LOCAL_CONFIG          ### inútil, não necessário
```

Sem sombra de dúvidas isso é muita informação para ocultar no arquivo `OSTYPE` — ele é quase tão flagrante quanto o estranho arquivo de 90 linhas do SuSE. Particularmente, a linha `DOMAIN` deveria estar no arquivo **.mc**, assim como deveriam estar várias das definições que influenciam o comportamento de sendmail. Muitas dessas linhas no arquivo `OSTYPE` não seriam necessárias se o Debian viesse com uma versão mais atual de **sendmail** e tirasse proveito dos padrões de **sendmail**.

www.snow666.blogspot.com
Snow666

A linha DONT_BLAME_SENDMAIL é curiosa. Ela joga fora completamente a segurança e é devastadora; a maioria das entradas não têm nada a ver com diretórios de origem. Esse arquivo também usa # para indicar uma linha de comentário, que não irá funcionar.

O arquivo **.mc** gerado pelo script de instalação é mostrado a seguir. Cada linha é o padrão do script, exceto pelo fato de termos solicitado smrsh e always_add_domain.

```
divert(0)
VERSIONID(`@(#)sendmail.mc        8.9.3-21 (Debian) 20000309')
OSTYPE(debian)
LOCAL_CONFIG                ### não é apropriado no meio de um arquivo mc
FEATURE(masquerade_envelope)
FEATURE(always_add_domain)
Cwlappie.toadranch.com      ### deveria estar no final abaixo de LOCAL_CONFIG
FEATURE(use_cw_file)
FEATURE(use_ct_file)
FEATURE(nouucp)
FEATURE(smrsh, `/usr/lib/sm.bin/smrsh')
MAILER_DEFINITIONS          ### não é apropriado
MAILER(local)
MAILER(smtp)

LOCAL_CONFIG
MASQUERADE_AS(lappie.toadranch.com) ### deveria estar acima de MAILERs
## Configurações personalizadas abaixo (serão preservadas)
```

Esse arquivo **.mc** apresenta uma série de problemas, inclusive instruções estranhas que não parecem se encaixar e instruções colocadas na ordem errada. É incrível que **m4** ainda seja capaz de lidar com isso e chegar a conclusões corretas.

Em geral, ocultar grande parte do arquivo de configuração no arquivo OSTYPE aumenta a complexidade e torna a depuração mais difícil à medida que você tenta descobrir como todos os recursos que você não ativou fazem parte, de forma mágica, de seu eventual arquivo de configuração. Por exemplo, o script de instalação optou por não incluir o recurso REDIRECT, mas ele ainda é listado no arquivo DOMAIN. Mesmo se disser não ao script durante o processo de configuração, você terá este recurso de qualquer maneira.

A versão atual do Debian, 3.0r1 vem com o **sendmail** 8.12.3. A seguir são mostrados os arquivos **debian.m4** e **sendmail.mc** conforme criados pelo script durante o processo de configuração.

```
divert(0)
`include(`/usr/share/sendmail/cf/debian/autoconf.m4')dnl')
VERSIONID(`$Id: debian.m4, v 8.12.3-6.6 2003-09-17 18:35:09 cowboy Exp $')
define(`confCF_VERSION', `Debian-6.6')dnl
dnl #
dnl # mudanças feitas aqui *devem* ser refletidas em parse_mc,update_db,debian.m4
dnl #
ifdef(`MAIL_SETTINGS_DIR', `dnl',
        `define(`MAIL_SETTINGS_DIR', `/etc/mail/')dnl')
dnl # suporta a migração do 8.9.3 para a converção de nomes de 8.10.0
dnl # Nota: isto aqui é muito tarde … precisa estar no sendmail .mc - no topo
ifdef(`_USE_ETC_MAIL_', `dnl', `define(`_USE_ETC_MAIL_')dnl')
dnl #
dnl # Define uma saudação-padrão
define(`confSMTP_LOGIN_MSG', `$j Sendmail $v/$Z; $b; (No UCE/UBE) $?{client_addr
}logging access from: ${client_name}(${client_resolve})-$_$.')
define(`confDEAD_LETTER_DROP', `/var/lib/sendmail/dead.letter')
define(`STATUS_FILE', `/var/lib/sendmail/sendmail.st')
define(`confHOST_STATUS_DIRECTORY',
        `/var/lib/sendmail/host_status')
define(`confCR_FILE', `-o ``MAIL_SETTINGS_DIR``relay-domains %[^\#]')
define(`confCT_FILE',      `MAIL_SETTINGS_DIR``trusted-users %[^\#]')
define(`confCW_FILE',      `MAIL_SETTINGS_DIR``local-host-names %[^\#]')
define(`confEBINDIR', `/usr/lib/sm.bin')
changequote([, ])
define([confMUST_QUOTE_CHARS], [.'])
```

www.snow666.blogspot.com
Snow666


```
changequote(`, ')
define(`confME_TOO', `True')
define(`confDEF_USER_ID', `mail:mail')
define(`LOCAL_MAILER_PATH',  `/usr/lib/sm.bin/sensible-mda')
define(`LOCAL_MAILER_ARGS',  `sensible-mda $g $u $h ${client_addr}')
define(`LOCAL_MAILER_FLAGS', `SPhnu9')
define(`LOCAL_SHELL_FLAGS',  `u9')
define(`PROCMAIL_MAILER_PATH', `/usr/bin/procmail')
define(`PROCMAIL_MAILER_ARGS', `procmail -m $h $f $u')
define(`USENET_MAILER_PATH', `/usr/bin/inews')
ifelse(sm_enable_ipv6, `yes',dnl
`ifelse(eval(sm_version_math >= 527360), `1',dnl
`define(`confBIND_OPTS', `+WorkAroundBrokenAAAA')dnl')')
ifelse(sm_enable_milter, `yes',dnl
define(`_FFR_MILTER')dnl')
ifelse(sm_enable_milter, `yes',dnl
`define(`_FFR_MAIL_MACRO')
```

O **sendmail.mc** criado pelo script de instalação é

```
divert(0)
define(`_USE_ETC_MAIL_')
include(`/usr/share/sendmail/cf/m4/cf.m4')
VERSIONID(`$Id: sendmail.mc, v 8.12.3-6.6 2003-09-17 18:35:09 cowboy Exp $')
OSTYPE(`debian')
DOMAIN(`debian-mta')
LOCAL_CONFIG
FEATURE(`masquerade_envelope')
LOCAL_CONFIG
Cwhost.dominio.com
FEATURE(`use_cw_file')
FEATURE(`use_ct_file')
FEATURE(`smrsh')
include(`/etc/mail/dialup.m4')
include(`/etc/mail/provider.m4')
MAILER_DEFINITIONS
MAILER(local)
MAILER(smtp)

LOCAL_CONFIG
## configurações personalizadas abaixo (serão preservados)
```

19.10 Recursos Relacionados com Spam no Sendmail

Spam é o jargão para correspondência indesejada, também conhecida como e-mail comercial não solicitado. Ele se tornou um problema sério, basicamente porque os remetentes (pelo menos nos Estados Unidos) não pagam por byte enviado, mas sim uma tarifa fixa pela conexão. Ou se pagam por byte, eles enviam uma única mensagem com milhares de receptores e retransmitindo-a através de uma outra máquina. A outra máquina paga o grande custo por byte e o spammer paga somente por uma cópia. Em vários países, usuários finais pagam por bytes recebidos e ficam muito irritados ao terem de pagar para receber spam.

O spam parece ser basicamente um problema nos Estados Unidos. A cultura de marketing norte-americana descobriu um filão de ouro e continua a explorá-lo.

Os ISPs nos Estados Unidos estão começando a sentir os efeitos do spam à medida que seu pessoal de suporte tem cada vez mais de lidar com ocorrências de abuso de spam originadas de seus clientes. Um ISP no Colorado com cerca de 150 clientes T1 (muitos deles ISPs) precisa de uma pessoa por meio-período somente para tratar de reclamações contra spams.

Sob o ponto de vista do pessoal de marketing, o spam funciona bem. As taxas de respostas são altas, os custos são baixos e a entrega é instantânea. Uma lista de 30 milhões de endereços de e-mail custa cerca de US$ 40.

Muitos spammers tentam parecer gentis sugerindo que você responda à mensagem deles com uma mensagem dizendo "remove" caso você queira ser removido da lista de e-mail deles. Embora até possam removê-lo da lista, você apenas acaba de confirmando que eles estão de posse de um endereço de e-mail atualizado e válido; este fato

www.snow666.blogspot.com
Snow666

pode levá-lo a ser incluído em outras listas. Os spammers também adoram bagunçar os cabeçalhos de seus e-mails numa tentativa de despistar de quem veio o e-mail e em qual máquina ele foi originado.

O pessoal que vende endereço de e-mail para spammers recentemente começou a adotar uma forma de ataque-dicionário para descobrir endereços desconhecidos. Começando com uma lista de sobrenomes comuns, o software de varredura acrescenta iniciais diferentes na esperança de atingir um endereço de e-mail válido. Para verificar os endereço, o software se conecta a servidores de e-mail em, digamos, 50 grandes ISPs e faz um VRFY ou RCPT em cada um dos zilhões de endereços.

Essa sondagem possui um impacto enorme sobre o servidor de e-mail e sobre sua capacidade de entregar a correspondência legítima para seus clientes. **sendmail** pode lidar com essa situação através da opção `PrivacyOption goaway` que é vista a partir da página 455. Mas os programas para spam mais inteligentes são muito robustos; se VRFY for bloqueado, eles tentam EXPN, e, se ambos falharem, tentam RCPT. Eles podem experimentar milhões de endereço dessa forma e jamais enviarem uma única mensagem — mas, certamente, mantêm o seu servidor de e-mail bem ocupado.

sendmail possui uma opção para frustrar tal comportamento, BAD_RCPT_THROTTLE. Se o número de endereços rejeitados no envelope de uma mensagem exceder o valor desta opção, **sendmail** começa a entrar em estado de hibernação por um segundo depois de cada comando RCPT rejeitado.

sendmail agregou alguns recursos muito interessantes para ajudar no controle de spam e também para ajudar a combater vírus de computadores ocasionais originados em mensagens de e-mails. Infelizmente, a maioria dos ISPs tem de repassar todos as mensagens; ou seja, tais recursos podem vir a ser muito draconianos para política de clientes (ou mais uma vez, talvez não sejam). Entretanto, os recursos podem ser usados com grande efeito no site do usuário final.

Há quatro tipos de recursos para controle de spam:

- Regras que controlam retransmissão promíscua ou de terceiros, que é o emprego de seu servidor de e-mail por um usuário fora do site para enviar e-mails para um outro usuário fora do site. Os spammers freqüentemente usam retransmissão numa tentativa de mascarar a verdadeira origem da mensagem e, portanto, evitar a detecção por parte de seus ISPs. Ela também permite que eles usem os *seus* ciclos e economizem os deles próprios. Isso é o principal.
- O banco de dados de acesso, que permite que os e-mails sejam filtrados por endereços de maneira bem parecida com um firewall para e-mails.
- Listas negras contendo retransmissores abertos e conhecidos sites amigos dos spams que **sendmail** pode consultar.
- Verificação de cabeçalhos e filtragem de e-mail de entrada por meio de uma interface de filtragem de e-mails genérica chamada **libmilter**. Ela permite varredura arbitrária dos cabeçalhos das mensagens *e conteúdo* e permite que você rejeite mensagens que coincidam com um determinado perfil.

Descreveremos esses novos recursos aqui e depois daremos uma olhada em alguns exemplos de spam que recebemos recentemente para ver como poderíamos ter ajustado nosso sistema de e-mail para reconhecê-los e rejeitá-los automaticamente.

Retransmissão

sendmail e outros agentes de transporte de e-mail aceitam e-mails que chegam, examinam os endereços do envelope, decidem para onde as mensagens devem ser encaminhadas e depois as repassam para um destino apropriado. Esste destino pode ser local ou um outro agente de transporte mais distante na cadeia de entrega. Quando uma mensagem que chega não tem nenhum recipiente local, o agente de transporte que a manipula é dito estar atuando como retransmissor.

Antes da versão 8.9 do **sendmail**, a retransmissão promíscua (também chamada de retransmissão aberta) era ativa por padrão. **sendmail** aceitaria qualquer mensagem apresentada a ele na porta 25 e faria o máximo para realizar a entrega. Era a política de vizinhança da Internet.

Infelizmente, os spammers começaram a abusar da retransmissão; eles a exploraram para disfarçar suas identidades e, mais importante, usar sua largura de banda e ciclos em vez de suas próprias. Atualmente é considerado muito ruim configurar o seu servidor de e-mails como um retransmissor aberto. Não obstante, muitos, mas muitos servidores mesmo, ainda estão configurados como retransmissores abertos (15% a 50% em algumas estimativas).

Somente hosts marcados com `RELAY` no banco de dados de acesso (veja a seção 19.10) ou listados em **/etc/mail/relay-domains** têm a permissão para submeter e-mails para retransmissão. A proporção dos retransmis-

www.snow666.blogspot.com
Snow666

sores abertos deve cair nos próximos anos como resultado dessa mudança no comportamento-padrão, juntamente com a ciência cada vez maior por parte do público e a ajuda de filtragem preventiva que usa as diversas listas de buracos negro.

Portanto, a retransmissão promíscua é negativa. Ao mesmo tempo, alguns tipos de retransmissão são úteis e legítimas. Como você poderia dizer quais mensagens devem ser retransmitidas e quais devem ser rejeitadas? A retransmissão é, na verdade, necessária somente em duas situações:

- *Quando o agente de transporte atua como gateway para hosts que não são atingíveis de outra maneira*; por exemplo, hosts que nem sempre estão ativos (hosts PPP, Windows PCs) e hosts virtuais. Nesta situação, todos os recipientes para os quais você quer retransmitir estão dentro do mesmo domínio.
- *Quando o agente de transporte é o servidor de e-mails de saída para outros hosts não tão inteligentes*. Neste caso, todos os nomes de hosts dos remetentes ou endereço IP serão locais (ou pelo menos enumeráveis).

Qualquer outra situação que pareça requerer retransmissão provavelmente é apenas uma indicação de projeto ruim (com a possível exceção do suporte para usuários móveis). Você pode prevenir o primeiro emprego de retransmissão (acima) designando um servidor centralizado para receber e-mails (com POP ou IMAP usado para acesso a clientes). O segundo caso deve sempre ser permitido, mas somente para os seus próprios hosts. Você pode verificar endereços IP ou nomes de host; nomes de host são mais fáceis de serem adulterados, mas **sendmail** verifica se eles não são forjados.

Embora **sendmail** venha com a opção de retransmissão desativada por padrão, vários recursos foram adicionados para reativá-la, tanto de uma maneira total como limitada e controlada. Esses recursos são listados a seguir para fins de integralidade, porém, nossa recomendação é que você seja cuidadoso em relação a abrir muito as coisas. A maioria dos sites não precisa de nenhum dos recursos realmente perigosos da segunda lista apresentada a seguir. O recurso `access_db`, visto na próxima seção, é a maneira mais segura de permitir retransmissão limitada.

- `FEATURE(`relay_entire_domain')` — permite retransmissão apenas para o seu domínio
- `RELAY_DOMAIN(``*dominio, ...*`')` — adiciona mais domínios para serem retransmitidos
- `RELAY_DOMAIN_FILE(`*nome-de-arquivo*`')` — idem, mas usa a lista de domínios de um arquivo
- `FEATURE(`relay_hosts_only')` — afeta `RELAY_DOMAIN`, `accessdb`

Você precisará abrir uma exceção caso use as designações `SMART_HOST` ou `MAIL_HUB` para direcionar e-mail através de uma determinada máquina servidora para e-mail. Esse servidor terá de ser configurado para retransmitir e-mails de hosts locais. Ele deve ser configurado com:

```
FEATURE(`relay_entire_domain')
```

Sites que realizam hospedagem virtual talvez também precisem de `RELAY_DOMAIN` para permitir a retransmissão para seus nomes virtuais, embora

```
FEATURE(`use_cw_file')
```

efetivamente abra retransmissão para estes domínios ou hosts.

Há algumas outras possibilidades, porém elas são cheias de problemas:

- `FEATURE('promiscuous_relay')` — permite retransmissão total; não use esta opção
- `FEATURE('relay_based_on_MX')` — retransmite para qualquer um que envie registros MX para você
- `FEATURE('loose_relay_check')` — permite endereçamento "percent hack"
- `FEATURE('relay_local_from')` — retransmissões baseadas no endereço From

O recurso `promiscuous_relay` permite retransmitir de qualquer site para qualquer outro site. Usá-la é um bilhete só de ida para as listas de buraco negro. *Não* use esse recurso numa máquina que possa ser atingida via Internet pública.

O recurso `relay_based_on_MX` é ruim, pois você não controla que sites têm permissão para apontar seus registros MX para você. Tipicamente, os únicos hosts que possuem um registro MX apontando para o seu servidor para e-mail são os seus próprios hosts, mas nada impede que outros sites alterem seus registros MX para apontar para você. Spammers normalmente não têm a capacidade de alterar registros MX, mas sites desconhecidos certamente teriam.

O recurso `loose_relay_check` permite a forma "% hack" (notação com o símbolo de porcentagem) de endereçamento (veja a página 388) que os spammers adoram usar.

Snow666

O recurso `relay_local_from` confia no endereço de remetente contido no envelope da mensagem e retransmite mensagens que pareçam ser provenientes de um endereço local. Obviamente, tanto o envelope quanto os cabeçalhos das mensagens de e-mail são fáceis de serem forjados, e os spammers são especializados em falsificações.

Caso considere ativar a retransmissão de alguma forma, consulte a documentação de **sendmail** em **cf/README** para garantir que você não se torne inadvertidamente um amigo dos spammers. Quando tiver terminado, faça com que um dos sites de verificação de retransmissão verifique se você não criou inadvertidamente um retransmissor aberto — experimente spam.abuse.net.

Há configurações incorretas na qual seu host poderia ser convencido a retransmitir endereços estranhos que usam a sintaxe de endereçamento UUCP de maneira abusiva. Apenas para se certificar, caso você não tenha conectividade UUCP, poderá usar

```
FEATURE(`nouucp', `reject')
```

para impedir tal possibilidade. O **sendmail** atual não tem como padrão suporte a qualquer uma das tecnologias de rede antigas como UUCP, BITNET ou DECnet.

Uma outra retransmissão comum é `LUSER_RELAY` para usuários locais que não existem. Ela é definida como padrão como

```
define(`LUSER_RELAY', `error:No such user')
```

Um site com **sendmail** mal configurado algumas vezes deixa vazar nomes de usuários locais não qualificados para a Internet (normalmente na linha Cc). Alguém que responda ao e-mail estará aparentemente respondendo para um usuário local que na verdade não existe. Essa retransmissão normalmente é chamada de "retransmissão de perdedor" e é direcionada para o agente de entrega `error`. Não há nenhuma necessidade de alterar essa configuração a não ser que queira retornar uma mensagem diferente ou implementar algum tipo de tratamento especial. Alguns sites redirecionam e-mails de "perdedor" para uma pessoa ou programa que realize concordância flexível, no caso de o remetente cometer um erro de digitação ou apenas ter o nome de login ligeiramente errado.

O banco de dados de acesso

sendmail inclui suporte a um banco de dados de acesso que você poderá usar para construir um firewall específico para e-mails para o seu site. Ele verifica e-mails provenientes do mundo externo e os rejeita caso provenha de usuários ou domínios específicos. Você também pode usar o banco de dados de acesso para especificar quais domínios uma máquina estará disposta a retransmitir.

O banco de dados de acesso é habilitado por meio da seguinte linha

```
FEATURE(`access_db', `tipo nome-de-arquivo')
```

Se *tipo* e *nome-de-arquivo* não forem especificados, o banco de dados usa como padrão o tipo `hash` (se forem usados bancos de dados DB — depende da configuração de `DATABASE_MAP_TYPE`) construído por meio do arquivo **/etc/mail/access**. Os bancos de dados DBM não usam o campo *tipo*. Como sempre, crie o banco de dados usando **makemap**:

```
# makemap hash /etc/mail/access < /etc/mail/access
```

O campo key do arquivo de acesso pode conter endereços de e-mail, nomes de usuário, nomes de domínios ou números de rede.

Por exemplo:

```
cyberspammer.com          550 Spam not accepted
okguy@cyberspammer.com    OK
badguy@aol.com            REJECT
sendmail.org              RELAY
128.32                    RELAY
170.201.180.16            REJECT
hotlivesex@               550 Spam not accepted
friend@                   550 You are not my friend!
```

A parte de valor deve conter um dos itens mostrados na Tabela 19.14. O valor `RELAY` é o mais permissivo; ele simplesmente aceita a mensagem e a encaminha para o seu destino final. `OK` aceita a mensagem porém não permitirá retransmissão. `REJECT` não aceitará a mensagem de forma alguma. `SKIP` permite que você abra exceções. Se você quiser, por exemplo, retransmitir e-mails para todos os hosts exceto dois num determinado domínio, poderá listar os dois hosts com a ação `SKIP` e, a seguir, listar o domínio com a ação `RELAY`. A ordem não importa.

Snow666

Tabela 19.14 Coisas que podem aparecer no campo de valor do banco de dados de acesso.

Valor	O que ele faz
OK	Aceita e-mail e o entrega normalmente
RELAY	Aceita e-mail como endereçado e o retransmite ao seu destino; habilita retransmissão por host
SKIP	Permite exceções a regras mais genéricas
REJECT	Rejeita o e-mail com uma mensagem de erro genérica
DISCARD	Descarta a mensagem silenciosamente
FRIEND	Para spams, usado pelo recurso delay-checks; se for coincidente, pula outras verificações de cabeçalho
HATER	Para spams, usado pelo recurso delay-checks; se for coincidente, aplica outras verificações de cabeçalho
xxx message	Retorna um erro; *xxx* tem de ser um código numérico da RFC821[a]
ERROR:*xxx message*	Mesmo que o anterior, mas claramente marcada com uma mensagem de erro
ERROR:*x.x.x message*	*x.x.x* tem de ser uma notificação de estado de entrega concordante com a RFC1893 (uma generalização do código de erro 550)

a. Por exemplo, 550 é o código de erro único.

O banco de dados precedente permite mensagens de okguy em cyberspammer.com, mas rejeitaria todas as demais mensagens de e-mail de cyberspammer.com com a mensagem de erro indicada. E-mails de sendmail.org ou então de 128.32.0.0/16 (rede da UC Berkeley) seriam retransmitidos. E-mails de badguy na AOL e do hotlivesex ou friend em qualquer domínio também seriam rejeitados.

Endereços IPv6 separados por dois-pontos podem ser usados também no lado esquerdo, porém teriam antes o prefixo "IPv6:". O símbolo @ após os nomes de usuários hotlivesex e friend é necessário para diferenciá-los dos nomes de domínios.

550 é um código de erro RFC821. Os códigos de erro RFC1893 (ou "mensagens de notificação de estado de entrega", como são chamados) são mais abrangentes. O primeiro dígito 4 indica um erro temporário; 5 significa um erro permanente. Listamos alguns na Tabela 19.15.

Tabela 19.15 Códigos de estado de entrega da RFC1893.

Temporário	Permanente	Significado
4.2.1	5.2.1	A caixa postal está desabilitada
4.2.2	5.2.2	A caixa postal está cheia
4.2.3	5.2.3	A mensagem é muito longa
4.2.4	5.2.4	Problema de expansão de lista
4.3.1	5.3.1	O sistema de e-mail está cheio
4.4.4	5.4.4	Incapaz de direcionar
4.4.5	5.4.5	Congestionamento de e-mails
4.7.*	5.7.*	Violação da política do site

Para controle ainda melhor, o campo key (lado esquerdo) pode conter as tags Connect, To, From e, nas versões 8.12 em diante, Spam para controlar a maneira pela qual o filtro é aplicado. Connect refere-se a informações de conexão tais como o endereço IP do cliente ou o nome de host do cliente. To e From referem-se aos endereços de envelope, não a cabeçalhos. A tag Spam permite exceções às regras globais através dos testes "spam friend" e "spam hater". Ela é habilitada com o recurso delay_checks:

```
FEATURE('delay_checks', 'friend')
FEATURE('delay_checks', 'hater')
```

O primeiro recurso pula outros conjuntos de regras que poderiam rejeitar a mensagem caso houvesse uma entrada concordante no access_db com FRIEND como o lado direito da associação. O segundo aplica os demais conjuntos de regras caso access_db seja HATER. Essas quatro tags fornecem um controle mais fino sobre retransmissão e rejeição de e-mails; elas podem ser usadas para prevalecerem sobre outras restrições também. Usuários individuais

Snow666

que reclamam de sua política contra spams válida para todo o site podem ser acomodados com as tags de spam FRIEND ou HATER.

Se uma dessas tags for usada, a pesquisa será tentada primeiro com a tag info e depois sem ela, para manter compatibilidade com versões de bancos de dados de acesso mais antigas.

Eis alguns exemplos:

```
From:spammer@algum.dominio      REJECT
To:bom.dominio                  RELAY
Connect:bom.dominio             OK
Spam:abuse@                     FRIEND
```

E-mails de spammer@algum.dominio seriam bloqueados, porém você ainda poderia enviar e-mails para este endereço, mesmo que ele se encontrasse na lista negra. E-mails serão retransmitidos para bom.dominio, porém não dele (supondo-se que a retransmissão tivesse sido desabilitada em algum ponto). As conexões em bom.dominio seriam permitidas mesmo que ele se encontrasse em uma das listas de rejeição segundo os DNS.

E-mails para abuse@dominiolocal passariam, mesmo de spammer@algum.dominio cujo e-mail teria sido rejeitado pela primeira linha do banco de dados de acesso.

Muitos sites usam um banco de dados de acesso para controlar spams ou políticas. Nossa máquina mestra de recebimento de e-mails no departamento de Ciências da Computação da Universidade do Colorado rejeita e-mails de mais de 500 spammers conhecidos identificados por endereços, domínios ou redes IP, tudo isto pelo emprego do recurso access_db.

Colocando usuários ou sites na lista negra

Se você possui hosts ou usuários locais para os quais deseja bloquear certos tipos de e-mails, use

```
FEATURE(`blacklist_recipients')
```

que suporta os seguintes tipos de entrada em seu arquivo de acesso:

```
To:nobody@                    550 Caixa postal desabilitada para este usuário
To:printer.meudominio.edu     550 Este host não aceita e-mail
To:usuario@host.meudominio.edu 550 Caixa postal desabilitada para este usuário
```

Essas linhas bloqueiam e-mails para o usuário nobody em qualquer host, para a impressora de host e para um endereço de um determinado usuário numa dada máquina. O emprego da tag To: permite que esses usuários enviem mensagens, mas não os deixa recebê-las; algumas impressoras possuem esta capacidade.

Para incluir as listas de buraco negro por meio de MAPS (Mail Abuse Prevention System; veja mail-abuse.org) — ou qualquer outra lista de bloqueio tipo DNS —, use o recurso dnsbl:

```
FEATURE(`dnsbl')
FEATURE(`enhdnsbl')
```

Esse recurso faz com que **sendmail** rejeite e-mails de qualquer site cujos endereços IP se encontram na Real-time Blackhole List de spammers conhecidos e mantida pelo MAPS. Outras listas catalogam spammers e sites discados conhecidos que executam retransmissores abertos. A versão otimizada do recurso dnsbl usa até cinco parâmetros e permite diferentes tipos de comportamento dependendo do valor retornado da pesquisa.

Essas listas negras são distribuídas através de um ajuste fino mais inteligente do sistema DNS; daí, o nome dnsbl. Por exemplo, um registro de recurso DNS especial da forma

```
Endereço-IP.blackholes.mail-abuse.org IN A 127.0.0.2
```

colocado no banco de dados DNS do domínio blackholes.mail-abuse.org bloquearia e-mails daquele host caso o recurso dnsbl estivesse habilitado (pois **sendmail** verificaria explicitamente para ver se um registro deste existia). *Endereço-IP* nesse exemplo é um endereço de host em sua forma de separação por quatro pontos com a ordem dos octetos invertida.

Você pode incluir o recurso dnsbl várias vezes para verificar diferentes listas de corruptores: acrescente apenas um segundo argumento para especificar o servidor de nomes de lista negra e um terceiro com a mensagem de erro que você gostaria que fosse retornada. Se o terceiro argumento fosse omitido, uma mensagem de erro fixa do banco de dados DNS contendo os registros seria retornada. Eis os exemplos de três listas: blackholes (o padrão), a lista de usuários discados dialups e a lista de retransmissores abertos relays.

```
FEATURE(`dnsbl', `blackholes.mail-abuse.org', `Rejeitado - see www.mail-abuse.org')
```

Snow666


```
FEATURE(`dnsbl', `dialups.mail-abuse.org', `Discados - see www.mail-abuse.org')
FEATURE(`dnsbl', `relays.mail-abuse.org', `Retransmissor - see www.mail-abuse.org')
```

O acesso às listas MAPS costumava ser gratuito, porém o serviço se tornou tão popular (com grandes provedores de e-mail solicitando milhões de registros por dia) que não é mais possível para o pessoal do MAPS atender às necessidades de toda a comunidade Internet sem qualquer suporte financeiro. Desde agosto de 2001, o serviço não é mais gratuito. As taxas variam de acordo com o tamanho e a natureza das organizações. Grandes usuários comerciais têm de pagar uma taxa de inscrição. Universidades, pequenas organizações e organizações sem fins lucrativos pagam menos e indivíduos ainda podem usar o serviço sem qualquer taxa. Veja www.mail-abuse.org para maiores detalhes.

O objetivo não é negar o serviço para ninguém, mas simplesmente fazer com que grandes usuários ajudem a pagá-lo. Os custos são reais: servidores maiores, mais largura de banda, contas de advogados já que os spammers processam (e sempre perdem), pessoal para manter as listas atualizadas etc. Agora todos os usuários têm de registrar e oficialmente optar por ele.

Os custos de pesquisa DNS extra nos servidores MAPS podem ser altos caso o seu site manipule grandes quantidades de e-mails. A maioria dos sites grandes mantém um servidor DNS secundário para as zonas MAPS para que as consultas possam ser resolvidas por um servidor local.

Verificação de cabeçalhos

Os spammers normalmente tentam ocultar suas identidades. A partir do **sendmail** 8.9, se o endereço From de envelope não estiver no formato *usuario@dominio.valido*, o e-mail será rejeitado. Esse comportamento pode ser renunciado pelos seguintes recursos:

```
FEATURE(`accept_unresolvable_domains')
FEATURE(`accept_unqualified_senders')
```

O primeiro desses recursos permite e-mails de domínios que não existem ou que não sejam determinados na árvore de atribuição de nomes DNS. O segundo permite endereços From contendo somente um nome de usuário sem nenhum trecho referente a host ou domínio. Não use nenhum desses recursos a menos que esteja protegido por um firewall e tenha somente dados DNS locais disponíveis aí. Caso esteja querendo ativar esses recursos, você deverá provavelmente pensar em reprojetar os seus ambientes DNS e **sendmail**. Exigir um endereço de remetente de envelope válido reduz significativamente os spams.

A verificação detalhada de cabeçalhos é um mecanismo anti-spam poderoso que faz uso da sintaxe de arquivo de configuração **sendmail** de baixo nível, que não é tratado aqui. Usando a verificação de cabeçalhos, **sendmail** pode procurar padrões especificados em cabeçalhos (por exemplo, "To:friend@public.com") e rejeitar mensagens antes de serem entregas às caixas postais de seus usuários.

A verificação de cabeçalhos também pode ser usada para reconhecer vírus transportados por e-mail desde que eles possuam uma linha de cabeçalho distinta. Por exemplo, o vírus Melissa de 1999 continha a linha de assunto: "Important Message From ...". Em poucas horas do vírus Melissa ter sido lançado e reconhecido, sendmail.com postou um conjunto de regras local para identificá-lo e descartá-lo. Quando a impressão digital de um vírus for distinta e fácil de expressar em regras de **sendmail**, sendmail.com normalmente posta uma correção para ele (tanto no site Web sendmail.com quanto no site sendmail.org) muito rapidamente.

Para uma amostra representativa das regras de filtragem contra spam e vírus, veja a configuração de **sendmail** para a máquina doméstica de Eric Allman, chamada knecht. Essa configuração é incluída na distribuição de **sendmail** como **cf/cf/knecht.mc**. Roube as regras de filtragem contra spam e acrescente-as ao final de seu arquivo **.mc**.

Ao analisarmos vários exemplos, encontramos regras de verificação de cabeçalhos para

- E-mails endereçados a qualquer usuário no domínio public.com
- E-mails endereçados a "friend" ou "you"
- E-mails com o cabeçalho X-Spanska, indicativo do worm Happy99
- E-mails com a linha de assunto "Important Message From ..." (o vírus Melissa)
- E-mails com a linha de assunto "all.net and Fred Cohen ..." (o vírus Papa)
- E-mails com a linha de assunto "ILOVEYOU" (o vírus iloveyou e variantes)
- Zilhões de mensagens de spam de marketing
- E-mails com um cabeçalho Outlook Express defeituoso (o worm SirCam)

www.snow666.blogspot.com
Snow666

Todas as regras de verificação vão sob instruções `LOCAL_CONFIG` e `LOCAL_RULESETS` no final do arquivo de configuração **.mc**. Com o auxílio do comando `divert` de **m4**, **sendmail** simplesmente sabe onde colocá-las no arquivo de configuração bruto.

Até certo ponto, qualquer combate a spams que você implementa bloqueia alguns spammers, mas abre um pouco mais a porta para os remanescentes. Use o agente de entrega `error` com uma mensagem de erro "usuario desconhecido" em vez do agente de entrega `discard`, pois muitos spammers limpam suas listas. Listas limpas são mais valiosas, de modo que você pode ser removido de algumas caso consiga interceptar o spam, filtrá-lo e responder com uma mensagem de erro.

Miltering: filtragem de e-mails

O **sendmail** versão 8.12 introduziu uma generalização da filtragem de cabeçalho que poderia se transformar numa ferramenta de combate a spams eficiente. Trata-se de uma API de filtragem de e-mails que as pessoas podem usar para desenvolver seus próprios programas de filtragem de e-mails. Esses programas de filtragem ficam entre **sendmail** e mensagens que chegam e podem ser usados para reconhecer o perfil de uma mensagem com spam ou vírus e descartá-la ou registrá-la em log (ou tomar qualquer outra providência que você julgue apropriada). Tanto conteúdo de mensagens quanto de metadados podem ser alvos da filtragem.

Miltering é, potencialmente, uma ferramenta poderosa tanto para combate a spams quanto para violar a privacidade dos usuários. Administradores que queiram saber exatamente que informações estão deixando a empresa via e-mail provavelmente sejam os primeiros a adotá-la. Miltering para e-mail que sai não é disponível na versão 8.12, mas se encontra na lista de prioridades a serem concretizadas.

A biblioteca miltering se chama **libmilter**. **sendmail** chama a filtragem de entrada por meio das diretivas de configuração `INPUT_MAIL_FILTER` ou `MAIL_FILTER` e controla a ação de miltering com opções chamadas `MILTER_MACROS_*` que permitem um controle mais refinado sobre os filtros aplicados a cada estágio da conversação SMTP.

Por exemplo, a linha:

```
INPUT_MAIL_FILTER('nome-do-filtro', 'S=mailer:/var/run/nome-do-filtro.socket')
```

passa cada mensagem que chega ao programa **/etc/mail/nome-do-filtro** pelo socket especificado no segundo argumento.

Para maiores informações, veja **libmilter/README** ou a documentação HTML no diretório **libmilter/docs** da distribuição **sendmail**. O arquivo **README** dá uma visão geral e um exemplo simples de um filtro que registra mensagens num arquivo. Os arquivos no diretório **docs** descrevem a interface da biblioteca e informa como usar as diversas chamadas para construir os seus próprios programas de filtragem. A filtragem de anexos feita no exemplo de configuração da máquina mestra de e-mail de sendmail.com (seção 19.9) também é um exemplo de miltering.

Manipulando spam

O combate ao spam pode ser uma tarefa difícil e frustrante. Passado um determinado ponto, é quase que totalmente fútil. Não se deixe seduzir pela idéia de capturar spammers individuais, muito embora vários passarão pelos seus escudos anti-spam. O tempo gasto analisando cabeçalhos de spam se preocupando com spammers é tempo perdido. Isso mesmo, está combatendo a boa luta, mas o tempo gasto nestas questões provavelmente não vai reduzir a quantidade de spam que chega a seu site.

Você pode agarrar spammers estacionários bem rapidamente classificando-os pelos seus ISPs, mas spammers tipo "bate-e-corre" que usam uma conta ISP uma única vez e depois a abandonam são muito mais difíceis de serem controlados. Se eles anunciam um site Web, então o site Web é o responsável; se for um número de telefone ou um endereço postal, é mais difícil de identificar o perpetrador, mas não impossível. Spammers móveis parecem basicamente estarem isentos de punição.

As diversas listas de buraco negro têm sido eficientes em parte no bloqueio a spam e reduziram dramaticamente o número de retransmissores abertos. Estar numa lista negra pode afetar seriamente os negócios, de modo que alguns ISPs e empresas são cuidadosos no policiamento de seus usuários. Nossa principal recomendação em relação a spam é que você use as medidas preventivas e listas negras mantidas publicamente que se encontram à disposição.

Alerte seus usuários para simplesmente eliminarem o spam que recebem. Muitas mensagens de spam contêm instruções sobre como aqueles que recebem a mensagem podem ser retirados da lista de endereços caso desejem. Se você seguir essas instruções, os spammers poderão retirá-lo da lista atual, mas o adicionarão imediatamente a várias outras listas com a nota "chegou a uma pessoa que leu a mensagem". O seu endereço de e-mail passa então a valer ainda mais.

Caso queira pegar carona na luta contra o spam, alguns sites Web podem ajudar. Dois sites impressionantes são mail-abuse.org e www.abuse.net. O site spamrecycle.com solicita que você envie a eles o seu spam; eles o enca-

Snow666

minham ao seu representante no Estado, que talvez opte por fazer algo politicamente em nível estadual. Esse site também possui um excelente conjunto de diretrizes para protegê-lo contra spam. O site analisa o spam e o utiliza para ajudar no melhoramento de filtros anti-spam.

Dois outros sites Web dignos de nota são spamcop.net e www.cauce.org. SpamCop possui ferramentas que ajudam a analisar cabeçalhos de e-mails e a determinar o verdadeiro remetente. O site www.cauce.org possui informações excelentes sobre legislação referentes a spam. O site Web www.orbs.org, que antigamente mantinha bancos de dados de retransmissores abertos, saiu da Internet em junho de 2001 e não deve mais ser usado.

Exemplos de spam

Embora não recomendemos analisar spam automaticamente, algumas vezes vale a pena saber como fazê-lo. Por exemplo, talvez você seja convocado para explicar porque o presidente de sua empresa recebeu uma solicitação de pornografia (e verificar que ela não proveio de um funcionário da empresa!).

Nas próximas páginas analisamos os cabeçalhos de alguns spams recentes. Estes exemplos ilustram quão difícil é determinar o verdadeiro remetente e quão fácil é falsificar cabeçalhos de e-mail. Em primeiro lugar, alguns pontos fundamentais:

- Cabeçalhos Received devem se encadear do início de uma mensagem até o seu final.
- Qualquer cabeçalho Received abaixo do cabeçalho Date provavelmente é falso.
- Anote qualquer cabeçalho Received em que os dois nomes de host não correspondam. Provavelmente o e-mail está sendo retransmitido por meio do primeiro host (o host entre parênteses é a origem real).
- Um cabeçalho Received com uma data antiga ou com uma data futura provavelmente é forjado.
- O trecho host do cabeçalho From deve ser concordante com o último cabeçalho Received a menos que a mensagem tenha sido enviada a uma lista de discussão.
- O domínio do cabeçalho Message-Id deve coincidir com o domínio do cabeçalho From a menos que o mascaramento no site do remetente altere as coisas.
- Verifique se os cabeçalhos Received mostram que a mensagem foi retransmitida por um host não relacionado. Algumas vezes isso é perfeito; por exemplo, uma empresa retransmitindo para a máquina doméstica de um funcionário ou vice-versa.
- Verifique todos os hosts listados para ter certeza de que eles existam no DNS. Use uma pesquisa MX, não uma pesquisa de registro A, pois firewalls locais ou DNS subdividido podem fazer com que a pesquisa A falhe.

Nosso primeiro exemplo é uma mensagem vendendo um CD contendo 10.000.000 de endereços de e-mail para futuros spammers. O CD com spam era interessante — ele garantia não conter endereços duplicados e nenhum endereço "envenenado" (presumidamente, endereços que enviam automaticamente o remetente a uma das listas de buraco negro).

Numeramos as linhas do cabeçalho para facilitar os comentários; os números realmente não se encontram lá.

```
1: From mrktnet77@kayak.msk.ru Thu Nov 4 22:10:48 1999
2: Received: from gaia.es ([195.55.166.66]) by xor.com (8.9.3/8.9.3) with ESMTP
      id WAA26343 for <evi@xor.com>; Thu, 4 Nov 1999 22:10:42 -0700 (MST)
3: From: mrktnet77@kayak.msk.ru
4: Received: from default by gaia.es (8.8.8+Sun/SMI-SVR4) id GAA03907; Fri, 5
   Nov 1999 06:31:10 -0100 (Etc/GMT)
5: Date: Fri, 5 Nov 1999 06:31:10 -0100 (Etc/GMT)
6: Received: from login_011556.wgukas.com (mail.wgukas.com [233.214.241.87])
      by (8.8.5/8.7.3) with SMTP id XAA01510 for fraklin321@thaxghklo.um.de;
      Thu, 4 November 1999 00:21:59 -0700 (EDT)
7: To: mrktnet77@kayak.msk.ru
8: Subject: Just Released! Millions CD Vol. 6A
9: Comments: Authenticated Sender is <user11556@wgukas.com>
10:Message-Id: 022021087226485974 56@sa_ghklo.um.de

/* Várias páginas de marketing eliminadas neste ponto */
*******************************************************
Do not reply to this message -
To be removed from future mailings:
mailto:greg1148@usa.net?Subject=Remove
*******************************************************
```

www.snow666.blogspot.com
Snow666

A linha 1 foi adicionada por **/bin/mail** durante a entrega local. O domínio msk.ru existe, mas o host kayak.msk.ru não; isto poderia ser legítimo se DNS subdividido estivesse em uso naquele site. A linha 2 é uma linha Received válida — ela é a única linha Received cuja precisão é garantida, pois foi adicionada pelo nosso próprio host (no caso, xor.com). A linha 3 é um cabeçalho From adicionado por **sendmail** durante o processo, pois a mensagem, originalmente, não possuía um.

A linha 4 é uma linha Received válida de um host bode expiatório não suspeito (gaia.es) rodando o sendmail 8.8, sob o qual a retransmissão é permitida por padrão (e que a Sun fornece desta maneira). O "from default" na linha 4 é uma pista de que a próxima linha Received é forjada.

A linha 6 é uma linha Received falsificada. Ela está abaixo da linha Date e estava faltando ou então já estava presente quando **sendmail** recebeu a mensagem. Além disso, o formato é incorreto, que normalmente é uma boa indicação de adulteração. O endereço IP 233.214.241.87 não possui nenhuma entrada DNS reversa. Em épocas passadas da Internet, voltada para o ambiente acadêmico e da pesquisa, isto também seria uma indicação de que algo estava errado. Hoje, muitos hosts não têm entradas DNS reversas.

A linha 7, a linha To, é incorreta. Os endereços dos receptores estavam somente no envelope.

A linha 9 pretende identificar o remetente autenticado o que, algumas vezes, é uma indicação da proveniência de uma mensagem. Esta implica que o remetente é de wgukas.com, mas este domínio não existe. Essa linha foi, na verdade, adicionada por um agente de usuário de e-mail num PC e, portanto, poderia muito bem ser forjada.

A linha 10 implica que a máquina remetente era realmente sa_ghklo.um.de, mas possuía o formato errado (faltando chaves em ângulo, < >) e, portanto, provavelmente forjada.

É impossível dizer de onde veio essa mensagem. Ela foi retransmitida através de gaia.es, provavelmente sem permissão. gaia.es ainda não se encontra na lista de buracos negros do MAPS, mas poderá acabar parando lá em breve. greg1148 poderia ser o próprio spammer ou ele poderia ser um usuário que reclamou em relação a um spam recebido anteriormente. No último caso, greg1148 assume o papel de vítima nessa mensagem e poderá receber centenas ou milhares de mensagens furiosas de pessoas pedindo para serem retirados da lista.

O corpo da mensagem solicitava que você telefonasse ou enviasse um fax do seu pedido para um número 800. É típico ter todas as informações necessárias para responder ao spammer e comprar seu produto no próprio corpo da mensagem. Note que o endereço da linha From é o mesmo do que o endereço da linha To; ambos provavelmente sejam forjados.

Um outro exemplo de spam desse mesmo dia ofertava-nos a possibilidade de nos tornarmos ricos caso enviássemos a eles, por fax, um cheque de US$ 40 até 15 de novembro, após o qual o preço subiria para US$ 195. Cópias em fax de um cheque são válidas legalmente? Ou os remetentes estão apenas interessados em obter o número de sua conta e sua assinatura para que eles mesmos imprimam seus próprios cheques? Proteja a sua identidade.

```
1: From jimdelno@apexmail.com Thu Nov 11 10:31:41 1999
2: Received: from saturn.globalcon.com (saturn.globalcon.com [209.5.99.8]) by
      xor.com (8.9.3/8.9.3) with ESMTP id KAA15479; Thu, 11 Nov 1999
      10:31:30 -0700 (MST)
3: Received: from hamilton ([168.191.61.20]) by saturn.globalcon.com
      (Post.Office MTA v3.1.2 release (P0205-101c) ID# 0-35881U1500L100S0)
      with SMTP id AAA148; Thu, 11 Nov 1999 12:33:24 -0500
4: Date: Thu, 11 Nov 1999 02:39:57 +0000
5: Subject: Free Information On "Debt Reduction!"
6: Message-Id: <yjsul.lnmqgaasnjymgqaac@hamilton>
7: From: F.Pepper@pmail.net
8: To: benfranklin@onehundred.net
```

A linha 2 é um cabeçalho Received válido. A linha 3 também é válida, mas **traceroute**s de xor.com, o destino, para hamilton (168.191.61.20) e saturn.globalcon.com (209.5.99.8) mostram que estes dois sites não têm nada a ver um com o outro. 168.191.61.20 se encontra na rede discada da Sprint e, a julgar pela indicação de fuso horário (que, infelizmente, é facilmente falsificável), poderia estar na Europa. 209.5.99.8 é uma empresa em Ontário, Canadá. saturn.globalcon.com provavelmente é um retransmissor aberto. Eles não estão usando **sendmail**, mas sim a versão 3.1.2 de Post.Office, agente de transporte de e-mails da Openwave.com (anteriormente, software.com)[36].

Na linha 4, o horário adicionado pelo agente de usuário do remetente é aproximadamente 2:00 da manhã na Europa, 2 horas antes dela ser recebida pela máquina saturn.globalcon.com. Talvez a lista de receptores fosse muito longa e tivesse levado 2 horas para processá-la. Ou quem sabe a mensagem foi composta fora do ar no PC do spam-

36. O Post.Office foi descontinuado pela Openwave (N.R.T.).

Snow666

mer e depois submetida à Internet mais tarde. A indicação de fuso horário (se não for falsa) apresenta 5 horas de diferença — a mesma diferença entre a Europa e a Costa Leste dos Estados Unidos.

Na linha 6, a porção host de Message-Id deveria ser um nome de domínio completamente qualificado, e não somente a parte local "hamilton". Esta provavelmente seja a máquina do spammer e seu nome provavelmente não seja hamilton. O trecho de Message-Id à esquerda do sinal @ normalmente é formado por números, mas também é permitido o emprego de letras, de modo que talvez esteja OK.

A linha 8 é claramente falsa. Os endereços dos verdadeiros receptores se encontravam apenas no envelope da mensagem e não aparecem em nenhum ponto dos cabeçalhos.

Esta mensagem poderia ser, na verdade, de F.Pepper@pmail.net. O nome de host pmail.net determina um endereço IP válido e **whois** informa que pmail.net é uma empresa britânica de telecomunicações.

Entre as 20 ou mais mensagens de spam que examinamos para esta seção, essa última sozinha poderia conter informações suficientes para se fazer uma reclamação razoável (ao hostmaster indicado no DNS para o endereço IP da linha 3 do cabeçalho). Jamais reclame diretamente com o spammer ou com o domínio do spammer.

SpamCop é um pacote de software que analisa cabeçalhos de e-mails e identifica que linhas são verdadeiras, quais provavelmente são forjadas e quais são totalmente falsas. Ele fornece aos usuários que submetem mensagens de spam através de e-mail ou da Web (spamcop.net) uma descrição detalhada dos cabeçalhos e informa que trechos devem ser examinados e quais não. Esse site também facilita o processo de submeter uma reclamação contra spam. A reclamação inclui todas as informações relevantes que você coletou da análise feita em seus cabeçalhos. SpamCop foi criado por Julian Haight.

Usamos SpamCop em nosso primeiro exemplo, que tentou nos vender um CD repleto de listas de endereços. Ele determinou que a linha Received de gaia.es estava OK, mas que o domínio wgukas.com era falso. A seguir ele determinou que gaia.es não estava de posse do endereço IP que a mensagem de e-mail informava e que o verdadeiro culpado provavelmente estivesse num site chamado ttd.net. Claramente a análise de SpamCop foi muito melhor do que a nossa e levou apenas alguns segundos.

Eis um pequeno trecho da análise do SpamCop para alguns spams:

```
Received: from sun1.cskwam.mil.pl (cskwam.mil.pl) [148.81.119.2] by
   mail1.es.net with smtp (Exim 1.81 #2) id 12oBHL-000494-00; Sat, 6 May
   2000 13:34:23 -0700
Possible spammer: 148.81.119.2
"nslookup cskwam.mil.pl" (checking ip) [show] ip not found; cskwam.mil.pl
   discarded as fake.
"dig cskwam.mil.pl mx" (digging for Mail eXchanger) [show] "nslookup
   cskwam.mil.pl" (checking ip) [show] cskwam.mil.pl not 148.81.119.2,
   discarded as fake.
"nslookup sun1.cskwam.mil.pl" (checking ip) [show] ip = 148.81.119.2
Taking name from IP
"nslookup 148.81.119.2" (getting name) [show] 148.81.119.2 =
   sun1.cskwam.mil.pl
"nslookup sun1.cskwam.mil.pl" (checking ip) [show] ip = 148.81.119.2
"nslookup 2.119.81.148.rbl.maps.vix.com." (checking ip) [show] not found
"nslookup 2.119.81.148.relays.orbs.org." (checking ip) [show] ip = 127.0.0.2
blocked by ORBS
Chain test:mail1.es.net =? mail1.es.net
Chain verified mail1.es.net = mail1.es.net
148.81.119.2 has already been sent to ORBS
Received line accepted
```

Cada uma das palavras [show] é um link para uma página Web da SpamCop. Eles mostram o verdadeiro comando que foi executado e a saída gerada por eles.

SpamAssassin

SpamAssassin é um filtro **procmail** muito eficiente na identificação de spams. Ele usa um sistema de pontuação para avaliar distorções de uma mensagem, raramente apresenta positivos falsos e captura todos os spams verdadeiros. Se uma mensagem obtiver muitos pontos (configurável), SpamAssassin a regrava numa pasta spam que você pode verificar quando estiver configurando e ajustando os parâmetros. Informações adicionais em spamassassin.taint.org.

Snow666

19.11 Segurança e o Sendmail

Com o crescimento explosivo da Internet, programas como o sendmail que aceitam entrada arbitrária fornecida por usuário e a entrega para usuários locais, arquivos ou shells têm, freqüentemente, fornecido uma avenida de ataque para os hackers. sendmail, juntamente com o DNS e até mesmo o IP, está flertando com a autenticação e criptografia como uma solução embutida para alguns dos problemas de segurança fundamentais.

A recente suavização das leis de exportação dos Estados Unidos em relação à criptografia permitiu que **sendmail** saísse de fábrica com ganchos embutidos para criptografia. As versões 8.11 e superiores suportam tanto criptografia quanto autenticação SMTP com TLS (Transport Layer Security, segurança de camada de transporte) (previamente conhecida como SSL, Secure Socket Layer). **sendmail** usa o termo TLS nesse contexto e o implementou como uma extensão, STARTTLS, ao protocolo SMTP. TLS trouxe com ele seis novas opções de configuração para arquivos de certificado e arquivos de chaves. Novas ações para concordâncias de banco de dados de acesso podem exigir que a autenticação tenha de ser bem-sucedida.

Nesta seção, descrevemos a evolução do modelo de permissões, propriedades e proteção de privacidade do **sendmail**. A seguir, discutimos brevemente o TLS e o SASL (Simple Authentication and Security Layer) e seus usos com o **sendmail**.

sendmail tem gradualmente enrijecido sua segurança ao longo do tempo e agora está muito chato em relação a permissões de arquivos antes de acreditar no conteúdo de, digamos, um arquivo **.forward** ou **aliases**. Embora esse enrijecimento da segurança geralmente seja bem-visto, algumas vezes é necessário relaxar as novas políticas rígidas. Para esse fim, **sendmail** introduziu a opção `DontBlameSendmail`,(NãoCulpeSendmail) assim chamada na esperança de que o nome sugerirá aos administradores de sistemas que aquilo que eles estão fazendo é considerado perigoso.

Esta opção possui vários valores possíveis — 55 na última contagem. O padrão é seguro. Para uma lista completa desses valores, veja **doc/op/op.ps** na distribuição **sendmail**. Os valores não são listados na segunda edição do livro da O'Reilly, **sendmail**, mas certamente estarão na terceira. Ou simplesmente deixe a opção configurada em `safe`.

Propriedades

Três contas de usuário são importantes no universo do **sendmail**: `DefaultUser`, `TrustedUser` e `RunAsUser`.

Como padrão, todos os agentes de entrega de **sendmail** são executados como `DefaultUser` a menos que os flags do agente de entrega especifiquem algo em contrário. Se um usuário mailnull, sendmail ou daemon existir no arquivo **/etc/passwd**, `DefaultUser` será este. Caso contrário, o seu padrão será UID 1 e GID 1. Recomendamos o uso da conta mailnull e de um grupo mailnull. Adicione-os a **/etc/passwd** com um asterisco como senha, nenhum shell válido, nenhum diretório de origem e um grupo-padrão mailnull. Você também terá de adicionar a entrada mailnull ao arquivo **/etc/group**. A conta mailnull não deve possuir qualquer arquivo. Se **sendmail** não estiver sendo executado como root, os agentes de entrega terão de ser setuid.

Se `RunAsUser` estiver ativo, **sendmail** ignorará o valor de `DefaultUser` e fará tudo como `RunAsUser`. Se você estiver executando **sendmail** como setgid (para smmsp), então o **sendmail** de submissão passará mensagens para o verdadeiro **sendmail** através de SMTP. O verdadeiro **sendmail** não possui o seu bit setuid ativo, mas ele é executado como root nos arquivos de inicialização.

`TrustedUser` de **sendmail** pode possuir mapas e arquivos de nomes alternativos. `TrustedUser` tem permissão para iniciar o daemon ou reconstruir o arquivo **aliases**. Esse recurso existe principalmente para oferecer suporte a interfaces GUI para **sendmail** que precisam fornecer controle administrativo limitado a certos usuários. Se você ativar `TrustedUser`, certifique-se de proteger a conta para a qual ele, aponta pois esta conta poderá ser facilmente explorada para ganhar acesso root. `TrustedUser` é diferente da classe `TRUSTED_USERS`, que determina quem pode reescrever a linha From de mensagens. [37]

`RunAsUser` é o UID que **sendmail** executa após abrir sua conexão de socket com a porta 25. Portas de numeração inferior a 1.024 podem ser abertas somente pelo superusuário; portanto, **sendmail** tem de ser executado inicialmente como root. Entretanto, após realizar essa operação, **sendmail** pode mudar para um UID diferente. Tal mudança reduz o risco de dano ou acesso caso **sendmail** seja enganado a fazer algo ruim. Não use o recurso `RunAsUser` em máquinas que suportem contas de usuário ou outros serviços; ele é concebido para uso em firewalls ou hosts bastiões somente.

---

37. O recurso `TRUSTED_USERS` é usado tipicamente para suportar software de listas de discussão. Por exemplo, se você usa o Majordomo, terá de acrescentar o usuário "majodormo" à classe `TRUSTED_USERS`. Os usuários root e daemon são os membros-padrão dessa classe.

www.snow666.blogspot.com
Snow666

Como padrão, **sendmail** não muda identidades e continua a ser executado como root. Se você alterar `RunAsUser` para algo diferente de root, terá de alterar várias outras coisas também. `RunAsUser` deve ser dono da fila de e-mails, ser capaz de ler todos os mapas e arquivos de inclusão, ser capaz de executar programas etc. Espere gastar algumas horas para descobrir todas as propriedades de arquivo e de diretórios que devem ser alteradas.

Permissões

As permissões sobre arquivos e diretórios são importantes para a segurança de **sendmail**. Use as configurações listadas na Tabela 19.16 para estar seguro.

Tabela 19.16 Proprietário e permissões para diretórios relacionados com sendmail.

Caminho	Proprietário	Modo	O que ele contém
/var/spool/clientmqueue	**smmsp**	770	Fila de e-mail para submissões iniciais[a]
/var/spool/mqueue	`RunAsUser`	700	Diretório de fila de e-mails
/, /var, /var/spool	**root**	755	Caminho para **mqueue**
/etc/mail/*	`TrustedUser`	644	Mapas, o arquivo de configuração, nomes alternativos
/etc/mail	`TrustedUser`	755	Diretório-pai para mapas
/etc	**root**	755	Caminho para o diretório **mail**

a. Versão 8.12 e posteriores.

sendmail não irá ler arquivos que possuem permissões frouxas (por exemplo, arquivos que podem ser gravados pelo grupo ou por todos ou que residam em diretórios que podem ser gravados pelo grupo ou por todos). Parte do rigor de **sendmail** em relação a propriedades e permissões foi motivada pelos sistemas operacionais que permitiam aos usuários transferirem a posse de seus arquivos com **chown** (estes derivados, em grande parte, do System V).

Os sistemas Linux, como padrão, possuem uma versão razoável de **chown** e não permitem a transferência de posse de arquivos. Entretanto, um `#ifdef` no código (`CAP_CHOWN`) pode ser utilizado para dar semântica de System V a **chown**. Você teria então que reconstruir o kernel. Mas esse comportamento é prejudicial; não force seu **chown** coerente do Linux a se comportar da maneira falha adotada pelo System V.

Em particular, **sendmail** é *muito* exigente em relação ao caminho completo a qualquer arquivo de nome alternativo ou arquivo de encaminhamento. Essa exigência algumas vezes colide com a maneira pela qual sites gostam de gerenciar nomes alternativos de listas de discussão Majordomo. Se, por exemplo, a lista Majordomo estiver em **/usr/local**, o caminho todo terá de ser admitido como confiável; nenhum componente pode ter permissão de gravação de grupo. Essa restrição torna mais difícil para o proprietário da lista gerenciar o arquivo de nomes alternativos. Para ver onde você se encontra em relação às idéias de **sendmail** sobre permissões, execute

```
# sendmail -v -bi
```

O flag **-bi** inicializa o banco de dados de nomes alternativos e o alerta sobre permissões inapropriadas.

Veja o Capítulo 16 para obter maiores informações sobre DNS.

sendmail não mais lerá um arquivo **.forward** que possui uma contagem de links maior do que 1 caso o caminho de diretório para ele seja inseguro (possua permissões frouxas). Essa regra pegou a Evi recentemente quando seu arquivo **.forward**, que tipicamente era um hard link tanto para **.forward.to.boulder** como para **.forward.to.sandiego**, falhou silenciosamente no encaminhamento de seu e-mail de um pequeno site no qual ela não recebia tantas mensagens. Levou meses para ela perceber e entender que "Eu jamais recebi o seu e-mail" era falha dela e não mais uma desculpa válida.

Você pode desativar várias das políticas restritivas de acesso a arquivos mencionadas anteriormente usando a opção `DontBlameSendmail`. Mas não faça isso.

E-mails mais seguros para arquivos e programas

Recomendamos que você use **smrsh** em vez de **/bin/sh** como shell do seu programa de entrega de e-mail e que use **mail.local** em vez de **/bin/mail** como seu agente de entrega local. Ambos os programas são inclusos na distribuição **sendmail**. Para incorporá-los em sua configuração, adicione as linhas

```
FEATURE(`smrsh', `caminho-para-smrsh')
FEATURE(`local_lmtp', `caminho-para-mail.local')
```

www.snow666.blogspot.com
Snow666

ao seu arquivo **.mc**. Se você omitir os caminhos explícitos, os comandos serão assumidos como residentes em **/usr/libexec**. Você pode usar a opção `confEBINDIR` de **sendmail** para alterar a localização-padrão dos binários para aquilo que desejar. O Red Hat fornece **/usr/sbin/smrsh**, porém nenhum **mail.local**. O SuSE fornece ambos em **/usr/lib/sendmail.d/bin**, e o Debian os guarda em **/usr/lib/sm.bin**.

smrsh é um shell restrito que executará somente os programas contidos num diretório (**/usr/adm/sm.bin** como padrão). **smrsh** ignora caminhos especificados por usuários e tenta localizar qualquer comando solicitado em seu próprio diretório conhecidamente seguro. **smrsh** também bloqueia o uso de certos metacaracteres de shell como "<", o símbolo de redirecionamento de entrada. São permitidos links simbólicos em **sm.bin**; portanto, você não precisa fazer cópias dos programas que você permite.[38]

O SuSE constrói **smrsh** para usar o diretório **/usr/lib/sendmail.d/bin**.

O **smrsh** do Debian é compilado para usar o diretório **/usr/lib/sm.bin**[39], que contém os programas **deliver**, **procmail** e **vacation** (todos links simbólicos). O link **deliver** é quebrado, e **procmail** não deveria estar lá. Ufa.

Eis alguns exemplos de comandos de shell e suas possíveis interpretações **smrsh**:

```
vacation eric              # executa /usr/adm/sm.bin/vacation eric
cat /etc/passwd            # rejeitado, cat não se encontra em sm.bin
vacation eric < /etc/passwd # rejeitado, < não é permitido
```

A opção `SafeFileEnvironment` de **sendmail** controla onde os arquivos podem ser gravados quando os e-mails são redirecionados a um arquivo por meio de um arquivo **aliases** ou **.forward**. Ela faz com que **sendmail** execute uma chamada de sistema **chroot**, fazendo com que a raiz do sistema de arquivos não seja mais **/**, mas sim **/safe**, ou seja lá qual for o caminho especificado na opção `SafeFileEnvironment`. Um nome alternativo que direcionou e-mail para o arquivo **/etc/passwd**, por exemplo, realmente seria gravado em **/safe/etc/passwd**.

A opção `SafeFileEnvironment` também protege arquivos de dispositivos, diretórios e outros arquivos especiais permitindo gravações somente em arquivos regulares. Além de aumentar a segurança, essa opção ajuda a amenizar os efeitos de erros cometidos pelos usuários. Alguns sites configuram a opção para **/home** para permitir acesso a diretórios de usuários embora mantendo arquivos de sistemas fora dos limites.

Os agentes de entrega também podem executar em um diretório ao qual foi aplicado **chroot**. Atualmente, essa opção deve ser especificada na definição do agente de entrega (mailer), mas ela deverá em breve ser configurável com **m4**.

Opções relacionadas com privacidade

sendmail também possui opções de privacidade que controlam

- O que o pessoal de fora pode determinar em relação ao seu site por meio do SMTP
- O que você exige do host na outra ponta de uma conexão SMTP
- Se seus usuários podem ver ou executar a fila de e-mails

A Tabela 19.17 lista os possíveis valores para as opções referentes à privacidade relativas à época em que este livro era escrito; veja o arquivo **doc/op/op.ps** na distribuição para informações atuais.

Recomendamos conservadorismo; use

```
define(`confPRIVACY_OPTIONS', ``goaway, authwarnings, restrictmailq,
   restrictqrun'')
```

em seu arquivo **.mc**. O valor-padrão de **sendmail** para as opções de privacidade é `authwarnings`; a linha acima reinicializaria este valor. Note os conjuntos de aspas duplas; algumas versões de **m4** as requerem para proteger as vírgulas na lista de valores de opção de privacidade. O Red Hat usa como padrão `authwarnings` e o SuSE `authwarnings`, `needmailhelo`, `novrfy`, `noexpn` e `noverb`.

---

38. Não coloque programas como **procmail** que podem disparar um shell em **sm.bin**. E não use **procmail** como o agente de entrega local, pois os usuários podem executar qualquer programa que desejarem de seus arquivos **~/.procmailrc**. Isso não é seguro.
39. A versão 8.12.3 do **sendmail** que vem com o Debian 3.0r1 utiliza o diretório **/etc/mail/smrsh** como base. Esse diretório contém apenas programas MDA, permitindo que o administrador do sistema faça suas mudanças, removendo ou acrescentando links simbólicos (N.R.T.).

www.snow666.blogspot.com
Snow666

Tabela 19.17 Valores da variável PrivacyOption.

Valor	Significado
`public`	Não realiza nenhum tipo de verificação de privacidade/segurança
`needmailhelo`	Requer HELO do SMTP (identifica host remoto)
`noexpn`	Não permite o comando EXPN do SMTP
`novrfy`	Não permite o comando VRFY do SMTP
`needexpnhelo`	Não expande endereços (EXPN) sem um HELO
`needvrfyhelo`	Não verifica endereços (VRFY) sem um HELO
`noverb`[a]	Não permite o modo prolixo para EXPN
`restrictmailq`	Permite somente que o grupo do diretório **mqueue** visualize a fila
`restrictqrun`	Permite somente que o proprietário do diretório **mqueue** execute a fila
restrictexpand	Restringe informações exibidas pelos flags **-bv** e **-v**[b]
`noetrn`[c]	Não permite execuções de filas assíncronas
`authwarnings`	Acrescenta o cabeçalho Authentication-Warning (este é o padrão)
`noreceipts`	Desativa a notificação de estado de entrega para recebimentos de retorno bem-sucedidos
`nobodyreturn`	Não retorna o corpo da mensagem num DSN
`goaway`	Desabilita todas as consultas de estado SMTP (EXPN, VRFY etc.)

a. Modo prolixo segue arquivos **.forward** quando é dado um comando EXPN e fornece maiores informações sobre os paradeiros de um e-mail de um usuário. Use `noverb` ou, melhor ainda, `noexpn` em qualquer máquina exposta ao mundo exterior.
b. Exceto se executado por root ou `TrustedUser`.
c. ETRN é comando do ESMTP concebido para emprego por hosts discados. Ele requer que a fila seja executada somente para mensagens para esse host.

Executando um sendmail onde se aplica chroot (para o verdadeiramente paranóico)

Se estiver preocupado em relação ao acesso que **sendmail** tem sobre o seu sistema de arquivos, poderá iniciá-lo numa cela onde se aplicou **chroot**. Crie um sistema de arquivos mínimo em sua cela, incluindo coisas como **/dev/null**, arquivos **/etc** essenciais (**passwd**, **group**, **resolv.conf**, **sendmail.cf**, quaisquer arquivos de mapeamento, **mail/***), as bibliotecas compartilhadas que **sendmail** precisa, o binário **sendmail**, o diretório de filas de e-mails e quaisquer arquivos de log. Provavelmente você terá de mexer na lista para fazer com que ela fique perfeita. Use o comando **chroot** para iniciar um **sendmail** deste tipo. Por exemplo:

```
# chroot /jail /usr/sbin/sendmail -bd -q30m
```

Ataques de recusa a serviços

Os ataques de recusa a serviços são difíceis de serem evitados, pois não há nenhuma maneira *a priori* que determine se uma mensagem é um ataque em vez de um e-mail válido. Os atacantes podem tentar várias coisas sórdidas, entre as quais inundar a porta SMTP com conexões inválidas, preencher partições de discos com mensagens gigantescas, entupir as conexões de saída e bombardeio de e-mail. **sendmail** possui alguns parâmetros de configuração que poderão ajudá-lo a reduzir ou limitar o impacto de um ataque de recusa a serviços, mas estes parâmetros poderão também interferir nas mensagens de e-mail legítimas. A nova biblioteca para filtragem de e-mails (com a versão 8.12) pode ajudar os administradores de sistemas a frustrar um ataque de recusa de serviço prolongado.

A opção `MaxDaemonChildren` limita o número de processos de **sendmail**. Ela impede que o sistema fique sobrecarregado com trabalho de **sendmail**, mas também permite que um atacante facilmente derrube um serviço SMTP. A opção `MaxMessageSize` pode ajudar a evitar a superlotação do diretório de filas de e-mails, mas se você configurá-la com um valor muito baixo, e-mails legítimos irão ricochetear. (Você pode comunicar o seu limite aos seus usuários de modo que eles não fiquem surpresos quando seus e-mails ricochetearem. De qualquer maneira recomendamos um limite relativamente alto, já que alguns e-mails legítimos são imensos.) A opção `ConnectionRateThrottle`, que limita o número de conexões permitidas por segundo, pode desacelerar as coisas um pouco. E, finalmente, configurar `MaxRcptsPerMessage`, que controla o número máximo de receptores permitidos numa única mensagem, poderá vir a ser útil.

www.snow666.blogspot.com
Snow666

sendmail sempre foi capaz de recusar conexões (opção `REFUSE_LA`) ou colocar e-mails em fila (`QUEUE_LA`) de acordo com a carga média de um sistema. Uma variação que foi introduzida na versão 8.12, `DELAY_LA`, permite que os e-mails continuem fluindo, mas numa velocidade reduzida. Veja a página 460 para obter maiores detalhes.

Apesar de todos estes "botões" para ajuste para ativar a proteção de seu sistema de e-mail, alguém que o esteja bombardeando com e-mails ainda interferirá em e-mails legítimos. O bombardeamento de e-mails pode ser bem sórdido.

A Universidade do Colorado fornece uma conta de e-mail para cada estudante (25.000), usando o **pine** como programa de correio eletrônico padrão. Alguns anos atrás, um estudante recém-contratado numa loja de informática da região foi convencido a dar uma cópia do arquivo de senhas a seu empregador. A empresa enviou então uma propaganda para todo o mundo que aparecia no arquivo de senhas, em lotes de aproximadamente 1.000 receptores por vez (o que gerou uma linha To muito longa).

O **pine** tinha sido compilado com o modo de resposta padrão configurado para responder a todos os receptores, bem como ao remetente. Muitos estudantes responderam a perguntas como: "Por que você me enviou este lixo?", e, obviamente, isto também foi enviado a cada uma das pessoas listadas na linha To. O resultado foi uma recusa de serviço total no servidor — para e-mails ou qualquer outro emprego. **sendmail** tomou todos os ciclos de CPU, a fila de e-mails ficou enorme e todo o trabalho útil ficou num estado de travamento. A única solução parecia ser colocar a máquina fora de linha, ir nas filas de e-mails e em cada caixa postal dos usuários e eliminar as mensagens prejudiciais. (Uma verificação de cabeçalho na linha Subject também poderia ter sido usada.)

Falsificações

Adulterar e-mails era, no passado, trivial. **sendmail** 8.10 e versões posteriores incluem autenticação SMTP que verifica a identidade da máquina remetente. Antes da versão 8.10, qualquer usuário poderia forjar e-mail para parecer como se ele viesse de seu domínio. A verificação de autenticação tem de ser ativada com a opção `AuthMechanisms`. Infelizmente, a autenticação de **sendmail** não é de ponta a ponta, mas sim entre servidores adjacentes. Se uma mensagem for manipulada por diversos servidores, a autenticação ajudará, mas não garantirá que a mensagem não foi forjada.

Da mesma forma, é possível personificar qualquer usuário em mensagens de e-mail. Tome cuidado se mensagens forem o veículo de autorização de sua organização para coisas como chaves, cartões de acesso e dinheiro. Você deve alertar os usuários administrativos sobre esse fato e sugerir que caso notem algum e-mail suspeito que pareça provir de uma pessoa com autoridade, eles devem verificar a validade da mensagem. Isso é mais verdade ainda se a mensagem solicitar que privilégios injustificáveis sejam dados a uma pessoa não usual. E-mail autorizando uma grande chave mestra para um estudante pode ser suspeito!

A opção de privacidade `authwarnings` sinaliza tentativas locais de falsificação adicionando um cabeçalho Authentication-Warning para e-mails de saída que parecem ser forjados. Entretanto, muitos agentes de usuário ocultam esse cabeçalho como padrão.

Se e-mail forjado estiver vindo de uma máquina que você controla, é possível, na verdade, fazer muito para frustrá-la. Você poderá usar o daemon **identd** para verificar o verdadeiro nome de login de um remetente. **sendmail** faz uma chamada de volta ao host remetente para solicitar ao **identd** que está sendo executado lá o nome de login do usuário que está enviando o e-mail. Se **identd** não estiver sendo executado no host remoto, **sendmail** não descobrirá nada. Se a máquina remota for uma estação de trabalho de um único usuário, seu proprietário poderá configurar **identd** para retornar uma resposta falsa. Mas se o host remoto for uma máquina multiusuário como aquela encontrada nos centros de computação de várias universidades, **identd** retornará o verdadeiro nome de login do usuário para **sendmail** para ser colocado no cabeçalho da mensagem.

Muitos sites não usam **identd**; ele normalmente é bloqueado por firewalls. **identd** somente é realmente útil dentro de um site, já que as máquinas que você não controla podem mentir. Num site grande, com usuários um tanto irresponsáveis (por exemplo, uma universidade), esse daemon é fantástico — mas ele também implicará perda de desempenho para **sendmail**.

Há vários anos, quando estávamos experimentando **identd** pela primeira vez, um estudante em nosso site ficou frustrado com os membros de sua equipe de projetos sênior. Ele tentou enviar e-mails para os seus colegas como se fosse o instrutor deles, informando-lhes que sabia que não estavam contribuindo o suficiente e que deveriam trabalhar de forma mais árdua. Infelizmente, ele cometeu um erro de sintaxe, e a mensagem ricocheteou para o instrutor. O uso por parte de **sendmail** do protocolo IDENT nos informou quem era ele. **sendmail** incluiu as seguintes linhas na mensagem ricocheteada:

```
The original message was received at Wed, 9 Mar 1994 14:51 -0700 from
    estudante@benji.Colorado.EDU [128.138.126.10]
```

Snow666

Mas os cabeçalhos da mensagem em si contavam uma outra história:

```
From: instructor@cs.Colorado.EDU
```

Moral da história: evite erros de sintaxe ao aprontar por aí. Nossa política sobre falsificação de e-mails fez com que o login desse estudante fosse desabilitado pelo resto do semestre que, na verdade, conseguiu exatamente o que o estudante queria. Ele não podia trabalhar no projeto e seus companheiros tiveram de trabalhar mais para suprir essa deficiência.

Consulte a seção 21.9 para obter maiores informações sobre PGP.

Privacidade de mensagens

A privacidade de mensagens basicamente não existe a menos que você use um pacote de criptografia externo como o PGP (Pretty Good Privacy), o GnuPG (GNU Privacy Guard) ou S/MIME. Por padrão, todas as mensagens são enviadas sem serem criptografadas. A criptografia de ponta a ponta requer suporte de agentes de usuário de e-mails.

Tanto o S/MIME quanto o PGP são documentados em RFCs, com o S/MIME já no caminho para se tornar um padrão. Entretanto, preferimos o PGP; ele é mais fácil de ser encontrado e foi concebido por um excelente especialista em criptografia, Phil Zimmermann, em quem confiamos. Esses padrões emergentes oferecem uma base para confidencialidade de e-mails, autenticação, garantia da integridade da mensagem e não-repúdio por parte da origem. A análise de tráfego ainda é possível, já que os cabeçalhos e o envelope são enviados em texto puro.

Informe a seus usuários que eles têm de gerar a sua própria criptografia caso queiram que seus e-mails sejam privados.

SASL: Simple Authentication and Security Layer

O **sendmail** 8.10 e versões posteriores oferecem suporte à autenticação SMTP definida na RFC2554. Ela se baseia no SASL (Simple Authentication and Security Layer) (RFC2222). O SASL é um sistema de segredos compartilhado que é tipicamente host-para-host; você tem de fazer arranjos explícitos para cada par de servidores que devem se autenticar mutuamente.

SASL é um mecanismo de autenticação genérico que pode ser integrado numa série de protocolos. Até então, **sendmail**, o **imapd** da Cyrus, o Netscape, o Outlook, o Mulberry e algumas versões do Eudora o utilizam. A estrutura SASL (ele é uma biblioteca) possui dois conceitos fundamentais: um identificador de autorização e um identificador de autenticação. Ele pode mapear esses conceitos a permissões em arquivos, senhas de contas, tíquetes Kerberos etc. O SASL contém tanto a parte de autenticação quanto a parte de criptografia, mas devido a limitações nas leis americanas de exportação, o **sendmail** usava somente a parte de autenticação na versão 8.10. Pelo fato de as leis americanas para exportação terem sido relaxadas no início do ano 2000, a parte referente à criptografia, que estava originalmente na lista de candidatos a estar presentes na versão comercial de **sendmail**, se tornou disponível para todo o mundo nas versões 8.11 e posteriores.

Para usar o SASL com **sendmail**, obtenha o Cyrus SASL do endereço a seguir:

ftp://ftp.andrew.cmu.edu/pub/cyrus-mail

TLS, um outro sistema de criptografia/autenticação é especificado na RFC2487. Ele é implementado no **sendmail** como uma extensão do SMTP chamada STARTTLS. Você pode até mesmo usar SASL e TLS juntos.

TLS é um pouco mais difícil de ser configurado e requer uma autoridade de certificação. Você pode pagar uma nota para a VeriSign para emitir seus certificados (chaves publicas assinadas identificando uma entidade) ou configurar sua própria autoridade de certificação. É usada autenticação rigorosa no lugar de um nome de host ou endereço IP como identificador de autorização para retransmissão de e-mails ou para aceitar uma conexão de um host em primeiro lugar. Uma entrada como

```
TLS_Srv:secure.example.com      ENCR:112
TLS_Clt:laptop.example.com      PERM+VERIFY:112
```

em `access_db` indica que STARTTLS se encontra em uso e que e-mails para o domínio secure.example.com devem ser criptografados com chaves criptográficas de pelo menos 112 bits. E-mails de um host no domínio laptop.example.com devem ser aceitos somente se o cliente tiver se autenticado.

Claus Assmann da Sendmail, Inc., produziu algumas páginas Web excelentes descrevendo como configurar SASL e TLS para uso com o **sendmail**:

www.sendmail.org/~ca/email/auth.html (SASL)
www.sendmail.org/~ca/email/starttls.html (TLS)

Essas instruções são particularmente úteis para o SASL, já que as instruções do próprio pacote são bem complicadas.

Snow666

Gregory Shapiro, também da Sendmail, Inc., criou alguns tutoriais excelentes sobre segurança e **sendmail**, que podem ser encontrados em www.sendmail.org/~gshapiro.

19.12 O Desempenho do SendMail

sendmail possui várias opções de configuração desenvolvidas para fins de melhorar o desempenho. Embora as tenhamos espalhado ao longo do capítulo, tentamos detalhar as mais importantes nesta seção. Há opções e recursos que você deve considerar caso cuide de um sistema de e-mails de grande volume (em ambas as direções). Na verdade, se você realmente precisa enviar 1.000.000 de mensagens por hora e não é um spammer, sua melhor opção é usar o lado comercial de **sendmail**, da Sendmail, Inc.

Modos de entrega

sendmail possui quatro modos básicos de entrega: em segundo plano, interativo, fila e protelação. Cada um deles representa uma relação de compromisso entre latência e vazão. O modo em segundo plano entrega e-mails imediatamente mas requer que **sendmail** crie um processo para fazê-lo. O modo interativo também entrega imediatamente, mas a entrega é feita pelo mesmo processo e faz com que o lado remoto aguarde resultados. O modo de fila coloca e-mails que chegam em fila, para entrega por um controlador de filas em algum momento no futuro. O modo de protelação é similar ao modo de fila, mas ele também protela todas as pesquisas de associação, de DNS, de nomes alternativos de encaminhamento. O modo interativo é raramente usado. O modo em segundo plano favorece baixa latência e os modos de protelação ou de filas aumentam a vazão. O modo de entrega é configurado com a opção `confDELIVERY_MODE` e o seu padrão é segundo plano.

Grupos de filas e divisão de envelopes

Os grupos de filas são um recurso novo do **sendmail** 8.12 que permite que você crie várias filas para mensagens que saem e controle individualmente os atributos de cada grupo de fila. Os grupos de fila podem conter um único diretório de filas ou vários diretórios. Se, por exemplo, o seu computador com Linux for o centralizador de e-mail para um ISP, você poderá definir um grupo de filas para os seus usuários discados e depois dar permissão aos usuários discados para iniciar uma execução de filas (usando o comando ETRN do SMTP) quando eles se conectarem para fazer o download dos e-mails. Os grupos de filas são usados com um recurso de subdivisão de envelope que permite que um envelope com vários receptores seja subdividido em grupos de filas. Esse recurso e o emprego de vários diretórios de filas por grupo de filas ajudam a amenizar os problemas de desempenho causados por termos um número excessivo de arquivos num único diretório de sistema de arquivos.

Quando uma mensagem adentra o sistema de e-mail, ela é atribuída a um ou mais grupos de filas. O grupo de filas para cada receptor é determinado de forma independente. Os envelopes são reescritos para corresponderem a atribuições do grupo de filas. Se forem usados vários diretórios de filas, as mensagens são atribuídas aleatoriamente às filas no grupo de filas correto.

Se um grupo de filas tiver um limite em relação ao número máximo de receptores por envelope, **sendmail** subdividirá o envelope da mensagem em vários envelopes menores que atendam aos parâmetros do grupo de filas.

Os grupos de filas são declarados com diretivas no arquivo **.mc**, mas são realmente configurados na linguagem do arquivo de configuração bruto pelo `LOCAL_RULESETS`, que não descrevemos neste livro. O exemplo a seguir o introduzirá no assunto caso queira usar grupos de filas para aumentar o desempenho ou para oferecer diferentes qualidades de serviço a diferentes destinos.

A Tabela 19.18 lista os atributos que podem ser especificados para um grupo de filas. Somente a primeira letra do nome do atributo precisa ser especificada quando o grupo de filas é definido.

Tabela 19.18 Atributos de grupos de filas.

Atributo	Significado
Flags	Principalmente para ajustes futuros; tem de ativar o flag **f** para ter vários executores de filas
Nice	Prioridade para este grupo de filas; diminui a prioridade de modo semelhante à chamada de sistema **nice**
Interval	Tempo a aguardar entre execuções de filas
Path	Caminho para o diretório de filas associado ao grupo de filas (obrigatório)
Runners	Número de processos **sendmail** a serem executados concorrentemente no grupo de filas
recipients	Número máximo de receptores por envelope

Snow666

Eis um exemplo que possui grupos de filas para e-mails locais, para e-mails para aol.com, para e-mails para outros sites remotos e uma fila-padrão para todos os demais e-mails. As linhas a seguir entram na parte comum do arquivo **.mc**:

```
dnl ##### - filas
QUEUE_GROUP(`local', `P=/var/spool/mqueue/local')
QUEUE_GROUP(`aol', `P=/var/spool/mqueue/aol, F=f, r=100')
QUEUE_GROUP(`remote', `P=/var/spool/mqueue/remote, F=f')
```

E depois no final do arquivo **.mc**:

```
LOCAL_RULESETS
Squeuegroup
R<$+>              $1
R$*@aol.com        $# aol
R$*@mydomain.com   $# local
R$*@$*             $# remote
R$*                $# mqueue
```

Nesse exemplo, especificamos um limite de 100 receptores por mensagem ao definirmos o grupo de filas AOL. Se uma mensagem de saída tivesse 10.000 receptores, dos quais 1.234 estivessem na AOL, a subdivisão de envelope colocaria 13 mensagens no grupo de filas `aol`, 12 de 100 receptores cada e 1 com os 34 restantes.

Para acelerar as coisas ainda mais, experimente subdivisão acelerada, que protela pesquisas MX durante o processo de ordenamento:

```
define('confFAST_SPLIT', '1')
```

Executores de filas

sendmail cria cópias de si mesmo para executar o transporte real das mensagens. Você pode controlar o número de cópias de **sendmail** que está sendo executado num dado momento e até mesmo quantas são alocadas a cada grupo de filas. Esse recurso permite que administradores de sistemas cheguem a um balanceamento entre **sendmail** e o sistema operacional em suas congestionadas máquinas centralizadoras de e-mail.

Três opções de **sendmail** controlam o número de daemons de executores de filas processando cada fila:

- A opção `MAX_DAEMON_CHILDREN` especifica o número total de cópias do daemon **sendmail** que são permitidas de serem executadas de uma só vez, incluindo aquelas filas em execução e as que aceitam mensagens que chegam.
- A opção `MAX_QUEUE_CHILDREN` configura o número máximo de executores de filas por vez.
- A opção `MAX_RUNNERS_PER_QUEUE` configura o limite de executores padrão por fila caso não seja especificado nenhum valor explícito no parâmetro `Runners=` (ou `R=`) na definição do grupo de filas.

Se você configurar valores que podem conflitar (por exemplo, um máximo de 50 executores de filas no total, porém 10 para a fila local, 30 para a fila mydomain e 50 para a fila AOL), **sendmail** colocará as filas em lote nos grupos de trabalho e fará alternância circular ("round robin") entre os grupos de trabalhos. Nesse exemplo, as filas local e mydomain seriam um grupo de filas e as filas AOL seriam um segundo grupo de trabalho. Se você optar por limites que certamente entrarão em conflito (por exemplo, max=50, porém AOL=100), **sendmail** usará `MAX_QUEUE_CHILDREN` como seu limite absoluto no número de executores de filas.

Controles de carga média

sendmail sempre foi capaz de recusar conexões ou colocar mensagens em filas em vez de entregá-las quando a carga média do sistema fica muito alta. Infelizmente, a carga média possui uma granularidade de apenas um minuto, portanto não é uma ferramenta muito efetiva para o controle dos recursos consumidos por **sendmail**. A nova primitiva `DELAY_LA` permite que configuremos um valor de carga média a partir do qual **sendmail** deve desacelerar; ele entrará em hibernação por um segundo entre comandos SMTP para as conexões atuais e antes de aceitar novas conexões. O valor padrão é 0, que desliga o mecanismo.

Mensagens na fila que não podem ser entregues

Mensagens que se encontram numa fila de e-mails, mas que não podem ser entregues, realmente podem acabar com o desempenho num servidor de e-mails. **sendmail** possui vários recursos que o ajudam na questão de mensa-

www.snow666.blogspot.com
Snow666

gens que não podem ser entregues. A mais eficiente é a opção FALLBACK_MX, que despacha uma mensagem para uma outra máquina caso ela não possa ser entregue na primeira tentativa. Esse recurso permite que sua máquina principal despache rapidamente mensagens para endereços válidos e desvie as mensagens problemáticas para uma segunda máquina de retaguarda. Um outro recurso útil é o diretório de estados de hosts, que armazena o estado dos hosts remotos através de execuções de filas.

A opção FALLBACK_MX é um grande trunfo para melhoria de desempenho para um site com grandes listas de discussão que invariavelmente contêm endereços que não podem, temporária ou permanentemente, serem atingidos. Para usá-la, você deve especificar o host que manipulará as mensagens proteladas. Por exemplo,

```
define(`confFALLBACK_MX', `mailbackup.xor.com')
```

encaminharia todas as mensagens que falhassem em sua primeira tentativa de entrega para o servidor central mailbackup.xor.com para processamento posterior. A partir da versão 8.12, podem existir várias máquinas de reserva caso vários registros MX sejam usados no DNS para os hosts designados.

Nas máquinas de reserva você pode usar a opção HOST_STATUS_DIRECTORY para ajudar em falhas múltiplas. Essa opção direciona **sendmail** a manter um arquivo de estados para cada host para o qual as mensagens são enviadas e a usar estas informações de estados para priorizar os hosts cada vez que a fila for executada. Essas informações de estados implementam caching negativo e permitem que informações sejam compartilhadas entre as execuções de filas. É um ganho de desempenho em servidores que manipulam listas de discussão com um grande número de endereços ruins, mas isto pode ser caro em termos de I/O de arquivos.

Eis um exemplo que usa o diretório **/var/spool/mqueue/.hoststat** (crie o diretório em primeiro lugar):

```
define(`confHOST_STATUS_DIRECTORY', `/var/spool/mqueue/.hoststat')
```

Se o diretório **.hoststat** for especificado com um caminho relativo, ele será armazenado abaixo do diretório de filas. **sendmail** cria a sua própria hierarquia interna de subdiretórios tomando como base o nome de host de destino.

Por exemplo, se e-mails para evi@anchor.cs.colorado.edu fossem falhar, as informações de estados iriam para o diretório **/var/spool/mqueue/.hoststat/edu./colorado./cs./** num arquivo chamado **anchor**, pois o host anchor possui um registro MX que aponta para ele mesmo como de maior prioridade. Se os registros MX do DNS tivessem direcionado mensagens para o host foo, então o nome de arquivo teria sido **foo** e não anchor.

Uma terceira melhoria em termos de desempenho para máquinas muito ativas envolve configurar um período de fila mínimo de modo que qualquer mensagem que não possa ser entregue na tentativa inicial possa ser colocada em fila e permanecer lá por um período mínimo entre tentativas. Essa técnica normalmente é acoplada a flags em linhas de comando que executam a fila mais freqüentemente (por exemplo, **-q5m**). Se um executor de filas ficar travado numa mensagem incorreta, um outro se iniciará em 5 minutos, aumentando o desempenho para as mensagens que podem ser entregues. A fila inteira é executada em lotes determinados por quais mensagens ficaram por lá pelo tempo mínimo exigido. Executar **sendmail** com os flags **-bd -q5m** e incluir a opção

```
define('confMIN-QUEUE_AGE', '27m')
```

no arquivo de configuração poderia resultar num sistema mais responsivo.

Ajustando o kernel

Caso pretenda usar um computador com Linux como servidor de e-mail de alto volume, você deve modificar vários parâmetros de configuração de rede do kernel e, talvez até mesmo construir um kernel personalizado (dependendo de sua configuração de hardware e carga esperada). Remova quaisquer drivers desnecessários para iniciar com um kernel mais enxuto que é suficiente para a sua configuração de hardware.

O kernel personalizado poderia incluir suporte a vários processadores caso a máquina host tivesse mais do que um processador (SMP). (Nós percebemos que para os verdadeiros *geeks* do Linux, esse comentário é análogo a um lembrete para não se esquecer de respirar. Mas já que as capacidades para construção de um kernel varia entre os usuários do Linux, ignoramos os comentários dos revisores e deixamos esse lembrete aqui.)

Para reinicializar os parâmetros da pilha de rede, use o comando **echo** do shell redirecionado para a variável apropriada no sistema de arquivos **/proc**. O Capítulo 13, *Redes TCP/IP*, contém uma descrição genérica desse procedimento na seção 13.10. A Tabela 19.19 mostra os parâmetros para alterar num servidor de e-mail de alto volume juntamente com seus valores sugeridos e padrões. Essas alterações provavelmente deveriam ser colocadas num script de shell que é executado no momento da inicialização e realiza os **echo**s correspondentes.

www.snow666.blogspot.com
Snow666

Tabela 19.19 Parâmetros de kernel a serem modificados em servidores de e-mail de alto volume.

Variável (relativa a /proc/sys)	Padrão	Sugerido
net/ipv4/tcp_fin_timeout	180	30
net/ipv4/tcp_keepalive_time	7200	1800
net/core/netdev_max_backlog	300	1024
fs/file_max	4096	16384
fs/inode_max	16384	65536

Você poderia, por exemplo, usar o comando

```
echo 30 > /proc/sys/net/ipv4/tcp_fin_timeout
```

para alterar o valor de timeout FIN do TCP.

19.13 Estatísticas, Testes e Depuração no SenDmail

sendmail pode coletar informações estatísticas sobre o número e tamanho de mensagens que ele manipulou. Você pode exibir esses dados com o comando **mailstats**, que organiza os dados por agente de entrega. A opção `STATUS_FILE` de **sendmail** (no arquivo `OSTYPE`) especifica o nome do arquivo no qual as informações estatísticas deveriam ser mantidas. A existência do arquivo especificado ativa a função de contabilidade.

A localização-padrão para o arquivo de informações estatísticas de **sendmail** é **/etc/mail/statistics**, mas alguns fornecedores chamam o arquivo **sendmail.st** e o colocam em **/var/log**.

Os totais exibidos pelos **mailstats** são cumulativos desde a criação do arquivo de estatísticas. Se você desejar estatísticas periódicas, você pode fazer um rodízio e reinicializar o arquivo por meio de **cron**.

Eis um exemplo:

```
% mailstats
Statistics from Wed Aug 1 02:13:30 2001
M     msgsfr      bytes_from     msgsto      bytes_to    msgsrej     msgsdis     Mailer
4         12            25K         63           455K          0           0      esmtp
7          0             0K         18            25K          0           0      relay
8         54           472K          0             0K          0           0      local
=============================================================================================
T         66           497K         81           480K          0           0
C         66                        81                         0
```

Se o arquivo de estatísticas de e-mails tiver permissão de leitura para todo o mundo, você não precisa ser root para executar **mailstats**.

Seis valores são mostrados: mensagens e kilobytes recebidos (`msgsfr`, `bytes_from`), mensagens e kilobytes enviados (`msgsto`, `bytes_to`), mensagens rejeitadas (`msgsrej`) e mensagens descartadas (`msgsdis`). A primeira coluna é um número identificando o agente de entrega e a última coluna lista o nome do agente de entrega. A linha `T` indica o total de mensagens e bytes e a linha `C` indica as conexões. Esses valores incluem tanto e-mails locais quanto retransmitidos.

Testando e depurando

As configurações baseadas em **m4** são, de certo modo, testadas previamente. Provavelmente não será necessário fazer depuração de baixo nível caso você as use. Uma coisa que os flags de depuração não são capazes de testar é o seu projeto. Enquanto pesquisávamos este capítulo, encontramos erros em vários dos projetos e arquivos de configuração que examinamos. Os erros variavam desde chamar um recurso sem a macro pré-requisito (por exemplo, usar `masquerade_envelope` sem ter ativado o mascaramento com `MASQUERADE_AS`) até conflito total entre o projeto da configuração de **sendmail** e o firewall que controlava se e sob quais condições as mensagens tinham permissão para entrar.

Você não pode projetar um sistema de e-mails num vácuo. Você tem de estar sincronizado com (ou pelo menos não estar em conflito) com seus registros MX do DNS e sua política de firewall.

sendmail fornece um dos conjuntos mais ricos do mundo em termos de ferramentas para depuração, com os flags de depuração que não são simples valores booleanos (verdadeiro ou falso) ou mesmo inteiros, mas são quantidades bidimensionais *x.y*, onde *x* seleciona o tópico e *y* seleciona a quantidade de informações a ser exibida. Um va-

Snow666

lor igual a 0 indica nenhuma depuração e 127 desperdiça muitas árvores se você imprimir a saída. Os tópicos variam de 0 a 99; atualmente são definidos cerca de 80.

O arquivo **sendmail/TRACEFLAGS** na distribuição lista os valores em uso e os arquivos e funções nas quais eles são usados. Todo o suporte de depuração se encontra em nível do arquivo de configuração bruta. Em muitos casos, é útil examinar o fonte de **sendmail** juntamente com a saída de depuração.

Se **sendmail** for chamado com um flag **-d**x.y, a saída da depuração vai para a tela (erro-padrão). A Tabela 19.20 ilustra valores importantes de *x* e alguns valores sugeridos para *y*. Seja cauteloso caso ative a depuração para um **sendmail** executado como daemon (**-bd**), pois a saída de depuração pode acabar misturada no diálogo SMTP e provocar falhas estranhas quando **sendmail** dialogar com hosts remotos.

Tabela 19.20 Tópicos de depuração.

Tópico	Significado e sugestões
0	Mostra flags de compilação e identidade de sistema (experimente y = 1 ou 10)
8	Mostra a resolução de nomes DNS (experimente y = 8)
11	Rastreia a entrega (mostra chamadas ao agente de entrega)
12	Mostra a conversão de nomes local-para-remoto
17	Lista hosts MX
21	Rastreia regras de reescrita (use y = 2 ou y = 12 para maiores detalhes)
27	Mostra nomes alternativos e encaminhamento (experimente y = 4)
44	Mostra tentativas de abertura de arquivos no caso de falhas (y = 4)
60	Mostra pesquisas de mapas de bancos de dados

checksendmail se encontra disponível em www.harker.com.

Gene Kim e Rob Kolstad escreveram um script Perl chamado **checksendmail** que chama **sendmail** em modo de teste de endereço com um arquivo de endereços de teste que você fornece. Ele compara os resultados àqueles esperados. Esse script lhe permite testar novas versões do arquivo de configuração em relação a um conjunto de testes dos endereços típicos de seu site para certificar que você não quebrou inadvertidamente qualquer coisa que costumava funcionar.

Entrega prolixa

Muitos agentes de usuário que chamam **sendmail** na linha de comandos aceitam um flag **-v**, que é passado para **sendmail** e faz com que ele exiba as etapas tomadas para entregar a mensagem. O exemplo a seguir usa o comando **mail**. Os termos em negrito foram digitados como entrada para o agente de usuário, e o restante é saída prolixa de **sendmail**.

```
% mail -v trent@toadranch.com
Subject: just testing, please ignore
hi
.
Cc:
trent@toadranch.com... Connecting to coyote.toadranch.com. via esmtp...
220 coyote.toadranch.com ESMTP Sendmail 8.11.0/8.11.0; Tue, 7 Aug 2001 20:
      08:51 -0600
>>> EHLO anchor.cs.colorado.edu
250-coyote.toadranch.com Hello anchor.cs.colorado.edu [128.138.242.1], pleased
      to meet you
250-ENHANCEDSTATUSCODES
250-EXPN
250-VERB
250-8BITMIME
250-SIZE
250-DSN
250-ONEX
250-ETRN
250-XUSR
250-AUTH DIGEST-MD5 CRAM-MD5
```

Snow666


```
250 HELP
>>> MAIL From:<evi@anchor.cs.colorado.edu> SIZE=65
250 2.1.0 <evi@anchor.cs.colorado.edu>... Sender ok
>>> RCPT To:<trent@toadranch.com>
250 2.1.5 <trent@toadranch.com>... Recipient ok
>>> DATA
354 Enter mail, end with "." on a line by itself
>>> .
250 2.0.0 f7828pi03229 Message accepted for delivery
trent@toadranch.com... Sent (f7828pi03229 Message accepted for delivery)
Closing connection to coyote.toadranch.com.
>>> QUIT
221 2.0.0 coyote.toadranch.com closing connection
```

O **sendmail** de **anchor** se conectou ao **sendmail** em toadranch.com. Cada máquina usou o protocolo ESMTP para negociar o intercâmbio da mensagem.

Falando em SMTP

Você pode fazer uso direto do SMTP ao depurar o sistema de e-mails. Para iniciar uma sessão SMTP, faça um **telnet** para a porta TCP 25 ou use **sendmail -bs**. Por padrão, essa é a porta na qual **sendmail** escuta quando executado em modo daemon (**-bd**); ele usa a porta 587 ao executar o agente de submissão de e-mails. A Tabela 19.21 mostra os comandos SMTP mais importantes.

Tabela 19.21 Comandos SMTP.

Comando	Função
HELO *hostname*	Identifica o host de conexão se estiver falando SMTP
EHLO *hostname*	Identifica o host de conexão se estiver falando ESMTP
MAIL From: *revpath*	Inicia uma transação de e-mail (remetente de envelope)
RCPT To: *fwdpath*[a]	Identifica receptor(es) de envelopes
VRFY *address*	Verifica se o endereço é válido (entregável)
EXPN *address*	Mostra a expansão de associações **.forward** e nomes alternativos
DATA	Inicia o corpo da mensagem[b]
QUIT	Finaliza o intercâmbio e fecha a conexão
RSET	Reinicializa o estado da conexão
HELP	Imprime uma síntese dos comandos SMTP

a. Pode haver vários comandos RCPT para uma mensagem.
b. Você encerra o corpo introduzindo um ponto sozinho em uma linha.

A linguagem completa possui somente 14 comandos, de modo que é bastante fácil aprendê-la e usá-la. Ela não distingue o uso de letras maiúsculas ou minúsculas. A especificação para SMTP pode ser encontrada na RFC821 (veja também a RFC1123). As RFCs 1869, 1870, 1891 e 1985 estendem o SMTP para ESMTP. Versões de próxima geração dessas referências existem na forma das RFCs 2821 e 2822; as mais antigas se tornarão obsoletas quando as novas forem adotadas formalmente como padrão.

A maioria dos agentes de transporte, inclusive **sendmail**, fala tanto SMTP como ESMTP; **smap** é a exceção isolada nos dias de hoje. Infelizmente, muitos computadores para firewalls que fornecem filtragem ativa não falam ESMTP.

Aqueles que falam ESMTP iniciam a conversação com o comando EHLO, e não HELO. Se o processo na outra ponta entende e responde com um OK, os participantes então negociam extensões suportadas e chegam a um mínimo denominador comum para a troca. Se for retornado um erro, então aquele que fala ESMTP retorna ao uso do SMTP.

Veja o Capítulo 11 para obter maiores informações sobre syslog.

Registro

sendmail usa syslog para registrar em log erros e mensagens de estados com o recurso de syslog "mail" e níveis "debug" através de "crit"; as mensagens são marcadas com a string "sendmail". Você pode alterar a string de registro "sendmail" com a opção de linha de comando **-L**, que é prática caso você esteja depurando uma cópia de **sendmail** enquanto outras cópias estão realizando tarefas de e-mail regulares.

Snow666

A opção `confLOG_LEVEL`, especificada na linha de comandos ou no arquivo de configuração, determina o nível de gravidade que **sendmail** usa como um limiar para registro. Valores altos do nível de log implicam níveis de gravidade baixos e fazem com que mais informações sejam colocadas em log.

Lembre-se de que uma mensagem logada em syslog num determinado nível é relatada a este nível e a todos aqueles acima dele. O arquivo **/etc/syslog.conf** determina o eventual destino de cada mensagem. Em sistemas Red Hat os logs de **sendmail** vão, por padrão, para **/var/log/maillog**. No SuSE a localização padrão é **/var/log/mail** e no Debian **/var/log/mail.log**. Não seria bom se as tentativas de padronização do Linux pudessem organizar estas diferenças aleatórias e aparentemente sem significado para que nossos scripts se tornassem mais portáveis de um sistema a outro?

A Tabela 19.22 dá uma idéia aproximada da associação entre níveis de log de **sendmail** e os níveis de gravidade de syslog.

Tabela 19.22 Níveis de log de sendmail *versus* níveis de syslog.

L	Níveis	L	Níveis
0	Nenhum registro	4	observação
1	alerta ou crítico	5–11	informação
2	crítico	>=12	depuração
3	erro ou alerta		

Um programa muito interessante chamado **mreport**, de Jason Armstrong, se encontra disponível em ftp://ftp.datrix.co.za/pub/mreport. Ele pode ser usado para sintetizar arquivos de log escritos por **sendmail**. Ele pode ser compilado sem problemas simplesmente usando-se **make** e **make install**. Eis um exemplo da saída gerada por **mreport** num sistema Red Hat:

```
# mreport -f -i /var/log/maillog -o mreport.out
[redhat.toadranch.com] [/var/log/maillog]

*  [  7] 592601  herb@yoherb.com                trent@toadranch.com
*  [  8] 505797  SNYDERGA@simon.rochester.edu   trent@toadranch.com
   [  1] 179386  steph@toadranch.com            bennettr@ci.boulder.co.us
   [  1]  65086  herb@yoherb.com                ned@xor.co
   [  7]  19029  evi@anchor.cs.colorado.edu     trent@toadranch.com
   [ 11]  17677  lunch-request@moose.org        trent@toadranch.com
   [  2]  16178  trent@toadranch.com            ned@camelspit.org
   [  3]  15229  reminders@yahoo-inc.com        herb@toadranch.com
   [  2]   4653  trent@toadranch.com            garth@cs.colorado.edu
   [  2]   1816  UNKNOWN                        trent@toadranch.com

. . . muitas linhas deletadas . . .
====================
Total Bytes        : 7876372
Number of Records  : 192
-------------
User Unknown       : 125
-------------
Host Name       : redhat.toadranch.com
Input File      : maillog
Output File     : mreport.out
First Record    : Aug 5 04:47:31
Last Record     : Aug 7 18:16:25
-------------
Time Taken      : 24317 µs
====================
```

Você tem de usar **sudo** ou ser root para executar **mreport** caso os seus arquivos de log de e-mail possam ser lidos apenas por root (como deveriam ser). Os flags e as opções estão documentados na página de manual para **mreport**. O flag **-f** nesse caso diz para agregar e ordenar por remetente; um flag análogo **-t** lhe permite ordenar por receptor. O argumento **-i** é o arquivo de entrada e **-o** o arquivo de saída.

www.snow666.blogspot.com
Snow666

Caso pretenda executar **mreport** regularmente, você deve informar os seus usuários. Eles podem achar uma invasão de privacidade os administradores de sistemas ficarem xeretando seus logs de e-mails através de uma ferramenta tão legal como esta.

19.14 O Sistema de E-Mail Exim

O Debian vem com o agente de transporte de e-mails Exim em vez de **sendmail**. Trata-se de um sistema mais simples com menos frescuras, mas também mais fácil de ser configurado para sites que não apresentam uma carga de e-mails pesada e não possuem um projeto de sistema de e-mails bizarro. A funcionalidade do Exim corresponde aos recursos mais comumente usados de **sendmail**.

Histórico

O Exim foi escrito em 1995 por Philip Hazel da Universidade de Cambridge e é distribuído sob a GNU Public License. Philip Hazel também é autor de um livro chamado *Exim: The Mail Transfer Agent* que foi publicado pela O'Reilly em julho de 2001. Transferimos a esse livro e à própria documentação do Exim em www.exim.org o local para que você obtenha detalhes sobre a configuração do Exim e fazemos apenas uma descrição rápida da versão do Debian aqui.

Exim no Debian

O Debian 3.0 inclui o Exim como seu programa de transporte de e-mails. Conforme vem de fábrica, o Debian simula o uso de **sendmail**; entretanto, **/usr/sbin/sendmail** e **/usr/lib/sendmail** são links para o binário do **exim** e **man sendmail** mostra a página de manual de **exim**. Os flags de linha de comando entendidos pelo **exim** coincidem com aqueles de **sendmail**, de modo que os agentes de usuário que chamam **sendmail** explicitamente com os flags **-bm** ou **-bs** ainda funcionam de forma adequada com o **exim** como agente de transporte de e-mails subjacente.

Como ocorre na família **sendmail**, o Debian dispõe de alguns comandos distintos que realizam funções de e-mail específicas. Estas são realmente implementadas por uma chamada a **exim** com certos flags de linhas de comando. A Tabela 19.23 ilustra os comportamentos e seus flags equivalentes.

Tabela 19.23 Utilitários de e-mail do Debian (com flags exim equivalentes).

Comando	Flag	Função
mailq	**-bp**	Mostra a fila de e-mails
rsmtp	**-bS**	Conexão SMTP em lote
rmail	**-i**	Para fins de compatibilidade com **smail**[a]
runq	**-q**	Executa a fila de e-mails

a. Aceita uma mensagem terminada com um ponto sozinho numa linha.

Configuração do Exim

O Exim contém três partes lógicas: diretores, roteadores e transportadores. Os diretores tratam dos endereços locais, isto é, endereços dentro do domínio de origem. Os roteadores manipulam endereços remotos, e os transportadores fazem a entrega em si.

O Exim é configurado de maneira muito parecida com o **smail3** ou Postfix com a linguagem de configuração assumindo a forma *palavra-chave = valor*. O Debian vem com um script em Perl que o ajuda a configurar o Exim perguntando-lhe algumas questões e contruindo o arquivo de configuração do Exim, **/etc/exim.conf**, de acordo com suas respostas.

Infelizmente, alguns dos valores-padrão o deixam com um sistema de e-mail configurado para imaginar que o seu nome de host é um domínio primeiro nível. Não precisa nem dizer; isso não funciona. Até mesmo mensagens que saem são afetadas, pois os endereços de remetente então passam a se parecer com *usuario@host;* devido a preocupações com spam, a maioria das instalações rejeita e-mails de remetentes cujos endereços de e-mail não são totalmente qualificados (isto é, eles rejeitam endereços que não contêm pelo menos um ponto). Para fazer com que o sistema de e-mail funcione corretamente, você tem de editar **exim.conf** manualmente ou executar **eximconfig** para fazer a verdadeira configuração inicial para você.

O arquivo **exim.conf** é muito bem documentado, com comentários precedendo cada variável que descrevem o que a variável faz, qual é o valor usual e que coisas (ruins) poderiam acontecer caso você falhasse em defini-la. O arquivo **exim.conf** padrão possui cerca de 100 variáveis definidas, mas isto é em grande parte linguagem-padrão que chama muito do comportamento de **sendmail** documentado nas seções anteriores.

Snow666

Em nosso sistema de testes tivemos de configurar somente quatro variáveis para fazer com que o sistema básico para e-mails funcionasse. Na verdade, duas destas variáveis se relacionam com questões de privacidade e não eram absolutamente exigidas. Eis as quatro variáveis, com nossos comentários ao lado:

```
qualify_domain = nome-do-domínio          ### por padrão configurada para nome de host desqualificado
local_domains = localhost:nome-do-domínio
smtp_verify = false                       ### o padrão é on, off desabilita SMTP VRFY
modemask = 002                            ### o padrão 022 assume um grupo para cada usuário
```

Não cobriremos com mais detalhes a configuração de Exim, mas iremos, em vez disso, indicar um excelente artigo do autor do Exim em sysadmin.oreilly.com/news.exim_0701.html[40], bem como a **/usr/doc/exim** (certifique-se de instalar o pacote **exim-doc,** utilizando o **apt-get**) e outras fontes mencionadas.

A seguir, descrevemos alguns recursos do Exim em termos de **sendmail** para que você possa comparar a funcionalidade dos dois sistemas e decidir se irá preferir instalar o verdadeiro **sendmail** em vez do Exim.

Similaridades entre o Exim e o sendmail

exim executa setuid para root de maneira muito parecida com que **sendmail** costumava fazer (antes da versão 8.12), de modo que você precisa ficar atento para estar atualizado em relação às correções de problemas de segurança. O conceito do Exim de usuários confiáveis coincide com aquele do **sendmail** e é usado basicamente para ajudar a facilitar o gerenciamento de listas de discussão, para as quais linhas From são rotineiramente reescritas. O Exim também lhe permite definir usuários administrativos que têm permissão para iniciar execuções de filas. O comando verify do SMTP (VRFY) é permitido por padrão.

exim tem de receber o sinal SIGHUP com o comando **kill** quando seu arquivo de configuração muda. Ele armazena o seu ID de processo em **/var/spool/exim/exim-daemon.pid**. Ele normalmente registra em log em arquivos contidos no diretório **/var/log/exim**, mas também pode usar syslog. Como padrão, o Debian vem com o **exim** registrando nos seus próprios arquivos e ignorando syslog, embora **syslog.conf** seja configurado com vários arquivos de log de e-mail que são vazios e entram num rodízio e são compactados todos os dias. (Arquivos vazios são maiores após a compactação do que antes dela!)

O Exim não entende endereços UUCP. Ele pode ser configurado a aceitar endereçamento com notação com símbolo de porcentagem, mas não faz isto por padrão.

O Exim permite que você encaminhe e-mails de saída para um host inteligente ("smart host") e filtre e-mails que chegam tanto no host quanto em nível de usuário. Ele suporta domínios virtuais e possui um banco de dados para nova tentativa, similar em funcionalidade ao diretório de estados de host de **sendmail**, para cuidar das entregas difíceis. Um mecanismo de filtragem que abranja todo o sistema (não por usuário) pode verificar a existência de anexos Microsoft, worms, vírus etc.

O Exim é capaz de usar listas de buraco negro em tempo real, mas possui o URL antigo para elas; você precisará mudar o caminho de vix.com para mail-abuse.org.

O Exim inclui um excelente recurso que não se encontra disponível em **sendmail**, ou seja, reconhecimento de loops em nomes alternativos e em encaminhamento e sua manipulação inteligente.

Finalmente, **man -k exim** nos leva a diversas ferramentas úteis para ajudar a manter o sistema de e-mail organizado.

19.15 Leitura Recomendada

Costales, Bryan e Eric Allman. *sendmail. 2. ed.* Sebastopol, CA: O'Reilly, 1997.
Este livro é o tomo definitivo — 1.000 páginas que valem a pena. Ele inclui um tutorial, bem como uma seção para referência muito completa. O livro é bem legível num estilo "abra uma página qualquer", o que consideramos ser um fator importante para um livro de referência. Ele também possui um excelente índice. Está na terceira edição. O tutorial será eliminado, mas o livro ainda continuará a ter por volta de 1.000 páginas!

Clayton, Richard. "Good Practice for Combating Unsolicited Bulk Email."
RIPE/Demon Internet. 2000. http://www.ripe.net/ripe/docs/ripe-206.html
Este documento visa aos ISPs. Ele possui uma grande quantidade de informações sobre políticas e alguns bons links para assuntos técnicos.

Schwartz, Alan; Paula Ferguson. *Managing Mailing Lists*. O'Reilly, 1998.
Este livro é uma boa referência sobre listas de discussão.

40. Esse é o URL atual: http://www.oreillynet.com/pub/a/oreilly/networking/news/exim_0701.html (N.R.T.).

Snow666

A página de manual para **sendmail** descreve seus argumentos de linha de comando. Veja *Sendmail: An Internetwork Mail Router*, de Eric Allman, para uma visão geral.
São apresentadas as instruções de instalação e uma boa descrição do arquivo de configuração em *Sendmail Installation and Operation Guide*, que pode ser encontrado no subdiretório **doc/op** da distribuição **sendmail**. Este documento é bem completo e, em conjunto com o arquiovo **README** no diretório **cf**, lhe dá uma boa visão sobre o funcionamento interno do **sendmail**.
www.sendmail.org, www.sendmail.org/~ca e www.sendmail.org/~gshapiro contêm documentos, HOWTOs e tutoriais relacionados com o **sendmail**.
A RFC2822, que torna obsoleta a RFC822, descreve a sintaxe das mensagens e endereços num sistema de e-mail em rede e a RFC1123 descreve as exigências do host. Estas são, num certo sentido, as especificações funcionais sobe as quais o **sendmail** foi construído.
A RFC2821, que torna obsoleta a RFC821, define o SMTP (Simple Mail Transport Protocol) e as RFCs 1869, 1870, 1891 e 1985 a estendem para o ESMTP.
A RFC974 descreve registros MX no DNS e seus relacionamentos com o direcionamento de e-mails.
Outras RFCs relacionadas com e-mail são:

- RFC1869 — Extensões de serviços SMTP
- RFCs 1891-1894 — Notificações de estados de entrega (mensagens ricocheteadas)
- RFC1985 — Filas remotas
- RFC2033 — LMTP (Local Mail Transport Protocol)
- RFC2034 — Códigos de erro SMTP
- RFC2045 — Extensões MIME
- RFC2476 — Especificações MSA (Mail Submission Agent)
- RFC2487 — SMTP seguro sobre TLS
- RFCs 2554 and 2222 — Autenticação SMTP e SASL

As RFCs 2821 (SMTP) e 2822 (Internet Message Format) organizam algumas das RFCs sobre e-mail mais comumente referidas na Internet; elas tornam obsoletas as RFCs 821, 822, 974 e 1869. As RFCs 2821 e 2822 foram publicadas pela primeira vez em abril de 2001 e são padrões propostos.
HAZEL, PHILIP. *Exim: The Mail Transfer Agent*. Sebastopol, CA: O'Reilly, 2001.
Documentação e informações sobre o Exim podem ser encontradas em www.exim.org.

19.16 EXERCÍCIOS

E19.1 Enumere brevemente as diferenças e as similaridades entre as opções `genericstable` e `virtusertable`. Em quais situações você usaria cada uma deles?

E19.2 Compare o emprego de **/etc/mail/aliases** com o emprego de um servidor LDAP para armazenar nomes alternativos de e-mails. Quais são as vantagens e as desvantagens de cada uma deles?

E19.3 Explique brevemente a diferença entre um MUA (Mail User Agent, agente de usuário de e-mail), um DA (Delivery Agent, agente de entrega) e um AA (Access Agent, agente de acesso). A seguir, explique a diferença entre um MTA (Mail Transport Agent, agente de transporte de e-mails) e um MSA (Mail Submission Agent, atente de submissão de e-mails).

E19.4 O que é **smrsh** e por que você deveria usá-lo em vez de **/bin/sh**? Se **smrsh** estiver sendo usado em seu site, que programas têm permissão para serem executados como o programa agente de entrega? Algum deles é perigosamente inseguro?

★ **E19.5** Faça uma breve descrição do seguinte cabeçalho de e-mail. Que caminho o e-mail percorreu? Para quem ele estava endereçado e para quem ele foi entregue? Quanto tempo levou para o e-mail ir do remetente ao destinatário?:

```
From clements@boulderlabs.com Fri Dec 28 17:06:57 2001
Return-Path: <clements@mail.boulderlabs.com>
Received: from boulder.Colorado.EDU (boulder.Colorado.EDU
     [128.138.240.1]) by ucsub.colorado.edu (8.11.6/8.11.2/ITS-5.0/student)
     with ESMTP idfBT06vF10618 for <hallcp@ucsub.Colorado.EDU>; Fri,
     28 Dec 2001 17:06:57-0700 (MST)
Received: from mail.boulderlabs.com (mail.boulderlabs.com
     [206.168.112.48]) by boulder.Colorado.EDU
```

Snow666


```
        (8.10.1/8.10.1/UnixOps+Hesiod (Boulder)) with ESMTP id
        fBTO6uL13184; Fri, 28 Dec 2001 17:06:56 -0700 (MST)
Received: from ath.boulderlabs.com (cpe-24-221-212-162.co.sprintbbd.net
        [24.221.212.162]) by mail.boulderlabs.com (8.11.6/8.11.6) with ESMTP
        id fBTO6oQ29214 for <booklist@boulderlabs.com>; Fri, 28 Dec 2001
        17:06:50 -0700 (MST) (envelope-from
        clements@mail.boulderlabs.com)
From: David Clements <clements@boulderlabs.com>
Received: (from clements@localhost) by ath.boulderlabs.com
        (8.11.6/8.11.4) id fBTO6ma01470 for booklist@boulderlabs.com; Fri,
        28 Dec 2001 17:06:48 -0700 (MST) (envelope-from clements)
Date: Fri, 28 Dec 2001 17:06:48 -0700 (MST)
Message-Id: <200112290006.fBTO6ma01470@ath.boulderlabs.com>
To: boolist@boulderlabs.com
Subject: Book Questions
```

★ **E19.6** Crie um pequeno arquivo **/etc/mail/aliases** que demonstre três tipos diferentes de nomes alternativos. Fale brevemente sobre o que faz cada uma das linhas e por que ele poderia ser útil.

★ **E19.7** Liste e explique o significado dos prefixos para arquivos no diretório de filas de e-mails. Por que é importante eliminar alguns deles, mas muito incorreto eliminar outros? Como alguns dos prefixos podem ser usados para depurar erros na configuração de **sendmail**? Observe **mailq** em seu servidor de e-mails. Algum problema no diretório? Há alguma mensagem sem arquivos de controle ou arquivos de controle sem mensagens? Qual é a mensagem mais antiga da fila? (Requer acesso root.)

★ **E19.8** Explique o propósito de cada uma das macros **m4** apresentadas a seguir. Se a macro inclui um arquivo, forneça uma breve descrição de qual deveria ser o conteúdo do arquivo.

a) VERSIONID
b) OSTYPE
c) DOMAIN
d) MAILER
e) FEATURE

★ **E19.9** Explique o significado de um registro MX. Por que os registros MX são importantes para entrega de e-mails? Dê um exemplo no qual um registro MX mal configurado poderia fazer com que o e-mail não pudesse ser entregue.

★ **E19.10** Quais são algumas das implicações de estar numa lista negra em MAPS ou numa lista de buraco negro de spam semelhante? Descreva algumas técnicas usadas para permanecer fora dessas listas.

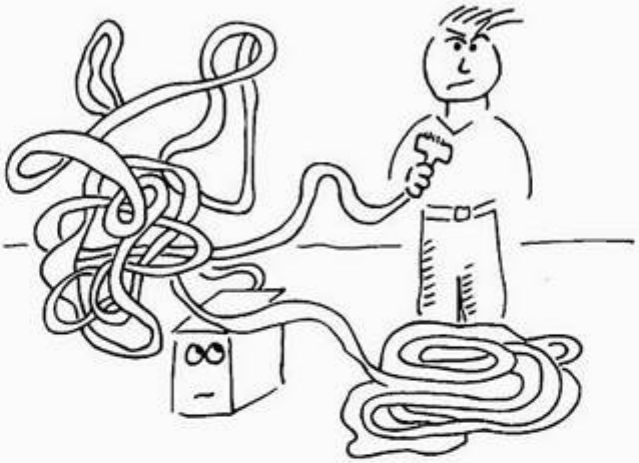
★ **E19.11** Se o seu site permite **procmail** (muito embora diríamos que ele não deveria) e se você tem permissão de seu grupo sysadmin local, configure o seu arquivo de configuração **procmail** pessoal para ilustrar como **procmail** pode comprometer a segurança.

★★ **E19.12** Explore a configuração **sendmail** atual em seu site. Quais são alguns dos recursos especiais de **sendmail** que estão sendo usados? Você é capaz de encontrar algum problema na configuração? Em que modos a configuração poderia ser melhorada?

★★ **E19.13** Localize um exemplo de spam em sua caixa postal e inspecione os cabeçalhos. Relate qualquer sinal de que a mensagem tenha sido forjada. A seguir execute algumas das ferramentas mencionadas neste capítulo, como SpamCop ou SpamAssassin, e relate suas descobertas. Como você faz para reconhecer cabeçalhos falsificados? Submeta o spam e suas conclusões sobre o remetente, a validade dos hosts listados e qualquer outra coisa mais que pareça estar inadequada.

www.snow666.blogspot.com
Snow666

20 Gerenciamento e Depuração de Redes



Pelo fato de as redes aumentarem o número de dependências entre as máquinas, elas tendem a aumentar os problemas. Como diria o ditado: "Rede é quando você não consegue ter nenhum trabalho terminado devido à falha de uma máquina da qual você jamais ouviu falar".

O gerenciamento de redes é a arte e a ciência de manter uma rede saudável. Geralmente abrange as seguintes tarefas:

- Detecção de falhas de redes, gateways e servidores críticos
- Esquemas para notificar a um administrador a existência de problemas
- Monitoramento geral, para balanceamento de carga e planejamento de expansões
- Documentação e visualização da rede
- Administração de dispositivos de rede a partir de um site central

Num único segmento de rede, em geral não vale a pena estabelecer procedimentos formais para gerenciamento da rede. Apenas teste completamente a rede após a instalação e verifique-a de vez em quando para ter certeza de que a carga não é excessiva. Quando a rede apresentar algum problema, corrija-o.

À medida que a rede for crescendo, os procedimentos de gerenciamento devem ser automatizados. Numa rede formada por várias sub-redes diferentes com switches (comutadores) ou roteadores, talvez você deva começar automatizando tarefas de gerenciamento com scripts de shell e programas simples. Caso tenha uma WAN ou uma rede local complexa, deve considerar a instalação de uma estação de gerenciamento de rede dedicada.

Em alguns casos, as necessidades de confiabilidade de sua organização ditarão a sofisticação de seu sistema de gerenciamento de redes. Um problema na rede pode fazer com que todo o trabalho seja paralisado. Caso o seu site não possa tolerar "interrupções", valerá a pena obter e instalar um sistema de gerenciamento de redes corporativas de alta capacidade.

Infelizmente, até mesmo o melhor sistema para gerenciamento de redes não é capaz de impedir todas as falhas. É crítico ter uma rede bem documentada e um pessoal altamente qualificado disponível para lidar com os colapsos inevitáveis.

Snow666

20.1 Diagnosticando uma Rede

Há várias ferramentas de qualidade para depurar uma rede na camada TCP/IP. A maioria delas gera informações de baixo nível, de modo que é necessário compreender as principais idéias do TCP/IP e roteamento para poder usar estas ferramentas de depuração.

Por outro lado, problemas de rede também podem surgir de questões referentes a protocolos de nível mais alto como DNS, NFS e HTTP. Talvez você queira ler os Capítulos 13, *Redes TCP/IP* e 14, *Roteamento*, antes de enfrentar este.

Nesta seção, começamos com alguma estratégia genérica de diagnóstico. A seguir vemos várias ferramentas essenciais, inclusive **ping**, **traceroute**, **netstat**, **tcpdump** e Ethereal.

Não discutiremos o comando **arp** neste capítulo, embora ele também seja uma ferramenta útil para depuração — veja a página 198 para obter maiores informações.

Quando sua rede está com problemas, é bem provável que você saia correndo para repará-la. Pare aí mesmo! É importante reservar um tempo e considerar como abordar o problema antes de mergulhar nele. O maior erro que você pode cometer é introduzir alterações mal planejadas numa rede já problemática.

Antes de tentar resolver o problema, considere os seguintes princípios:

- Faça uma alteração de cada vez e teste cada uma delas para ter certeza de que tenha o efeito desejado. Desfaça quaisquer alterações que produzam um efeito indesejado.
- Documente a situação em que ela se encontrava antes de você se envolver e documente cada alteração que você faça no meio do caminho.
- Inicie numa "ponta" do sistema ou rede e vá trabalhando nos componentes críticos até atingir o problema. Por exemplo, talvez você possa começar analisando a configuração de rede num cliente, percorrer as conexões físicas, investigar o hardware da rede e, por fim, verificar as conexões físicas do servidor e configurações de software.
- Comunique-se regularmente. A maioria dos problemas em uma rede envolve ou afeta um grande número de pessoas diferentes: usuários, ISPs, administradores de sistemas, engenheiros de telecomunicações, administradores de redes etc. Uma comunicação clara e consistente impedirá que você acabe obstruindo os esforços de outros em solucionar o problema.
- Trabalhe em equipe. Anos de experiência demonstram que as pessoas cometem menos erros estúpidos se elas tiverem um companheiro ajudando.
- Use as camadas de rede para negociar o problema. Comece pelo "alto" ou por "baixo" e vá percorrendo a pilha do protocolo.

Este último ponto merece ser discutido um pouco mais. Conforme descrito na página 182, a arquitetura do TCP/IP define várias camadas de abstração nas quais os componentes da rede podem funcionar. Por exemplo, o HTTP depende do TCP, o TCP depende do IP, o IP depende do protocolo Ethernet e o protocolo Ethernet depende da integridade do cabo de rede. Você pode reduzir dramaticamente a quantidade de tempo gasto depurando um problema se você descobrir primeiramente qual camada está se comportando de maneira incorreta.

Faça as seguintes perguntas a si próprio à medida que você sobe (ou desce) pela pilha:

- Você tem conexão física e uma indicação (led) de link?
- Sua interface está configurada de maneira apropriada?
- O DNS está configurado apropriadamente?[1]
- Suas tabelas ARP mostram outros hosts?
- É possível fazer um ping para o endereço de localhost (127.0.0.1)?
- É possível fazer um ping para outros hosts locais pelo endereço IP?
- É possível fazer um ping para outros hosts locais pelo nome de host?
- É possível fazer um ping para outros hosts em uma outra rede?
- Comandos de alto nível como **telnet** e **ssh** funcionam?

Assim que tiver identificado onde se encontra o problema, retorne um passo e considere o efeito que testes subseqüentes e possíveis correções terão sobre outros serviços e hosts.

---

1. Se sua máquina travar no momento da inicialização, inicializar muito lentamente ou ficar travada em conexões **telnet** de entrada, o DNS deve ser seu principal suspeito.

www.snow666.blogspot.com
Snow666

20.2 PING: VERIFIQUE SE UM HOST ESTÁ FUNCIONANDO

O comando **ping** é embaraçosamente simples, porém em muitas situações é tudo de que você precisa. Ele envia um pacote ICMP ECHO_REQUEST a um host de destino e espera para ver se o host responde de volta. Apesar de sua simplicidade, o comando **ping** é um dos burros de carga na depuração de redes.

Podemos usar **ping** para verificar o estado de hosts individuais e testar segmentos da rede. As tabelas de rede, redes físicas e gateways estão envolvidos no processamento de um ping, de modo que a rede tem de estar mais ou menos funcionando para um comando **ping** ser bem-sucedido. Se **ping** não funcionar, pode estar quase certo de que nada mais sofisticado funcionará. Entretanto, essa regra não se aplica a redes que bloqueiam solicitações de eco ICMP através de um firewall. Certifique-se de que um firewall não esteja interferindo em sua depuração antes de chegar à conclusão de que o host de destino está ignorando um **ping**. Talvez você possa considerar a possibilidade de desabilitar o firewall por um curto espaço de tempo para facilitar a depuração.

Cada fornecedor dispõe de um comando **ping**. A maioria das versões de **ping** executa um loop infinito a menos que o argumento de contagem de pacotes seja dado. Assim que tiver satisfeito com o resultado de ping, digite o caractere de interrupção (normalmente <Control-C>) para sair.

Eis um exemplo:

```
% ping beast
PING beast (10.1.1.46): 56 data bytes

64 bytes from 10.1.1.46: icmp_seq=0 ttl=255 time=0.808 ms
64 bytes from 10.1.1.46: icmp_seq=1 ttl=255 time=0.400 ms
64 bytes from 10.1.1.46: icmp_seq=2 ttl=255 time=0.390 ms
^C
--- beast ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.390/0.533/0.808/0.195 ms
```

A saída para beast mostra o endereço IP do host, o número de seqüência ICMP de cada pacote de resposta e o tempo de viagem de ida e volta. A coisa mais óbvia que essa saída nos diz é que o servidor beast está no ar e conectado à rede.

Numa rede sem problemas, **ping** permite que você determine se um host está inativo ou não. Por outro lado, quando um host remoto sabidamente está ativo e funcionando, **ping** pode lhe dar informações úteis sobre a saúde da rede. Os pacotes **ping** são direcionados por mecanismos IP usuais e uma viagem de ida e volta bem-sucedida significa que todas as redes e gateways que se encontram entre a fonte e o destino estão funcionando corretamente, pelo menos numa primeira abordagem.

O número de seqüência ICMP é uma informação particularmente valiosa. Descontinuidades na seqüência indicam pacotes perdidos. Apesar do fato de o IP não garantir a entrega dos pacotes, uma rede saudável deve perder muito poucos deles. A rede pode parecer funcionar corretamente, mas será muito mais lenta do que deveria ser, não somente devido aos pacotes retransmitidos, como também pelo custo adicional do protocolo necessário para detectá-los e gerenciá-los.

Para descobrir a causa do desaparecimento de pacotes, execute primeiramente **traceroute** (veja a próxima seção) para conhecer a rota que os pacotes estão tomando em direção ao host desejado. A seguir, efetue um comando **ping** nos gateways intermediários em seqüência para descobrir que link está perdendo pacotes. Para capturar o problema, você precisa enviar um número estatisticamente significativo de pacotes. A falha na rede geralmente recai sobre o link entre o último gateway sobre o qual você consegue aplicar um **ping** sem perdas de pacotes significativas e o gateway acima dele.

O tempo de viagem de ida e volta informado por **ping** lhe dá uma visão do desempenho global de uma rota através de uma rede. Pequenas variações no tempo de viagem de ida e volta normalmente não indicam problemas. Os pacotes podem ocasionalmente atrasar em cerca de décimos ou centésimos de milissegundos sem motivo aparente; esta é simplesmente a forma de funcionamento do IP e do Linux. Você deve esperar observar um tempo de viagem de ida e volta razoavelmente consistente para a maioria dos pacotes, com lapsos ocasionais. Muitos dos roteadores atuais implementam respostas a pacotes ICMP proporcionalmente limitadas, significando que um roteador pode atrasar para responder ao seu **ping** caso ele já esteja lidando com muito tráfego ICMP.

O programa **ping** lhe permite enviar pacotes de solicitação de qualquer tamanho. Usando um pacote maior do que o MTU da rede (1.500 bytes para redes Ethernet), você poderá forçar a ocorrência de fragmentação. Essa prática o ajudará a identificar erros de mídia ou outros problemas de baixo nível como aqueles de uma rede ATM congestionada. É surpreendentemente o número de problemas que você pode descobrir desta maneira.

Snow666

Podemos especificar o tamanho do pacote desejado em bytes com o flag **-s**:

```
# ping -s 1500 cuinfo.cornell.edu
```

Use o comando **ping** tendo em mente as seguintes limitações. Primeiramente, é difícil distinguir uma falha de rede de uma falha de um servidor somente com o comando **ping**. Um ping falho apenas lhe informa que *algo* está errado.

Em segundo lugar, um **ping** não garante muito em relação ao estado da máquina de destino. Os pacotes de solicitação são manipulados dentro do protocolo IP e não requerem que um processo de servidor esteja rodando no host testado. Uma resposta garante somente que uma máquina está ligada, e não "kernel panic" (isto é, travamento de kernel). Você precisará de métodos de nível mais alto para verificar a disponibilidade de serviços como o HTTP e o DNS.

20.3 TRACEROUTE: RASTREIA PACOTES IP

traceroute, criado por Van Jacobson, permite que você descubra a seqüência de gateways através da qual um pacote IP viaja para alcançar o seu destino.

A sintaxe é simplesmente

```
traceroute nome-do-host
```

Há uma série de opções, e a maioria delas não é importante no dia-a-dia. Como sempre, *nome-do-host* pode ser especificado tanto simbólica quanto numericamente. A saída é simplesmente uma lista de hosts, iniciando com o primeiro gateway e terminando no destino.

Por exemplo, um **traceroute** proveniente do host jaguar para o host drevil gera a seguinte saída:

```
% traceroute drevil
traceroute to drevil (192.225.55.137), 30 hops max, 38 byte packets
 1 xor-gw2 (192.108.21.254) 0.840 ms 0.693 ms 0.671 ms
 2 xor-gw4 (192.225.56.10) 4.642 ms 4.582 ms 4.674 ms
 3 drevil (192.225.55.137) 7.959 ms 5.949 ms 5.908 ms
```

Com base nessa saída podemos dizer que jaguar se encontra exatamente a três saltos de distância de drevil e podemos ver que gateways estão envolvidos na conexão. O tempo de viagem de ida e volta para cada gateway também é mostrado — são medidas e exibidas três amostras para cada salto. Um **traceroute** típico entre hosts Internet pode incluir dez ou vinte saltos.

traceroute funciona configurando-se o campo TTL (Time To Live, na realidade "tempo de sobrevivência da contagem de saltos") de um pacote de saída para um número artificialmente baixo. À medida que pacotes chegam num gateway, seus TTLs são diminuídos. Quando um gateway diminui o valor TTL para ficar igual a 0, ele descarta o pacote e envia uma mensagem ICMP "time exceeded" (tempo excedido) para o host de onde proveio o comando.

Refira-se à página 310 para obter maiores informações sobre pesquisas DNS.

Os primeiros pacotes **traceroute** possuem seu TTL configurado em 1. O primeiro gateway a ver um pacote desses (xor-gw2 neste caso) determina que o TTL foi excedido e notifica jaguar sobre o pacote descartado enviando uma mensagem ICMP de volta. O endereço IP no cabeçalho do pacote de erro identifica o gateway; **traceroute** pesquisa esse endereço no DNS para descobrir o nome de host do gateway.

Para identificar o gateway de segundo salto, **traceroute** envia uma segunda rodada de pacotes com campos TTL configurados em 2. O primeiro gateway direciona os pacotes e diminui o TTL de uma unidade. No segundo gateway, os pacotes são então descartados e as mensagens de erro ICMP geradas como antes. Este processo continua até que o TTL seja igual ao número de saltos para o host de destino e os pacotes atinjam seus destinos.

A maioria dos roteadores envia suas mensagens ICMP por meio da interface "mais próxima" ao seu host. Se você executar **traceroute** percorrendo o caminho contrário por meio do host de destino, provavelmente verá endereços IP diferentes sendo usados para identificar o mesmo conjunto de roteadores. Talvez você também veja caminhos completamente diferentes; esta configuração é conhecida como "roteamento assimétrico".

Como **traceroute** envia três pacotes para cada valor do campo TTL, você pode, algumas vezes, observar um resultado interessante. Se um gateway interventor multiplexar tráfego através de várias rotas, os pacotes poderão ser retornados por hosts diferentes; nesse caso, **traceroute** simplesmente imprimirá todos eles.

Vejamos um exemplo mais interessante de um host em colorado.edu para xor.com:

```
rupertsberg% traceroute xor.com
traceroute: Warning: xor.com has multiple addresses; using 192.225.33.1
traceroute to xor.com (192.225.33.1), 30 hops max, 40 byte packets
 1 cs-gw3-faculty.cs.colorado.edu (128.138.236.3) 1.362 ms 2.144 ms 2.76 ms
 2 cs-gw-dmz.cs.colorado.edu (128.138.243.193) 2.720 ms 4.378 ms 5.052 ms
```

Snow666


```
3 engr-cs.Colorado.EDU (128.138.80.141) 5.587 ms 2.454 ms 2.773 ms
4 hut-engr.Colorado.EDU (128.138.80.201) 2.743 ms 5.643 ms 2.772 ms
5 cuatm-gw.Colorado.EDU (128.138.80.2) 5.587 ms 2.784 ms 2.777 ms
6 204.131.62.6 (204.131.62.6) 5.585 ms 3.464 ms 2.761 ms
7 border-from-BRAN.coop.net (199.45.134.81) 5.593 ms 6.433 ms 5.521 ms
8 core-gw-eth-2-5.coop.net (199.45.137.14) 53.806 ms * 19.202 ms
9 xor.com (192.225.33.1) 16.838 ms 15.972 ms 11.204 ms
```

Essa saída mostra que os pacotes têm de passar por cinco de nossos gateways internos antes de deixarem a rede colorado.edu (de cs-gw3-faculty a cuatm-gw). O gateway no próximo salto na rede BRAN (204.131.62.6) não possui um nome no DNS. Após dois saltos em coop.net, chegamos a xor.com.

No salto 8, vemos um asterisco no lugar de um dos tempos de viagem de ida e volta. Essa notação indica que nenhuma resposta (pacote de erro) foi recebida em resposta à sondagem. Nesse caso, a causa provavelmente é congestionamento, mas esta não é a única possibilidade. **traceroute** depende de pacotes ICMP de baixa prioridade, que muitos roteadores são inteligentes o bastante para descartar em preferência a tráfego "verdadeiro". Você não deve entrar em pânico caso haja um pequeno número de asteriscos.

Se você vir asteriscos em todos os campos de tempo de viagem de ida e volta para um dado gateway, nenhuma mensagem "time exceeded" (tempo excedido) está chegando daquela máquina. Quem sabe o gateway esteja simplesmente fora do ar. Algumas vezes, um gateway será configurado para descartar silenciosamente pacotes com TTLs vencidos. Nesse caso, você ainda será capaz de ver através do host silencioso em direção aos gateways mais distantes. Uma outra possibilidade é que os pacotes de erros do gateway sejam lentos para retornarem e que **traceroute** tenha deixado de esperar por eles no momento em que eles chegam.

Alguns firewalls bloqueiam totalmente mensagens ICMP de "tempo esgotado". Se houver algum desses firewalls ao longo do caminho, você não receberá informações sobre qualquer um dos gateways acima dele. Entretanto, você ainda pode determinar o número total de saltos ao destino, pois os pacotes de sondagem irão eventualmente percorrer todo o caminho. Da mesma forma, alguns firewalls podem bloquear os datagramas UDP vindos de fora que **traceroute** envia para disparar respostas ICMP. Este problema faz com que **traceroute** não gere nenhuma informação útil.

Um link lento não indica necessariamente um defeito. Algumas redes físicas possuem uma latência alta natural. Lentidão também pode ser um sinal de congestionamento na rede receptora, especialmente se a rede usa uma tecnologia CSMA/CD que faz repetidas tentativas de transmitir um pacote (Ethernet é um exemplo). Tempos de viagem de dia e volta inconsistentes suportariam tal hipótese, já que as colisões aumentam a aleatoriedade do comportamento da rede.

Algumas vezes você poderá ver a notação `!N` em vez de um asterisco ou tempo de viagem de ida e volta. Isso indica que o gateway atual retornou um erro "network unreachable" (rede inatingível), significando que ele não sabe como direcionar o seu pacote. Outras possibilidades incluem `!H` para "host unreachable" (host inatingível) e `!P` para "protocol unreachable" (protocolo inatingível). Um gateway que gera alguma dessas mensagens de erro normalmente será o último salto que você poderá alcançar. Esse host normalmente apresenta um problema de roteamento (possivelmente provocado por um link interrompido): suas rotas estáticas estão erradas ou os protocolos dinâmicos falharam em propagar uma rota útil ao destino.

Caso pareça que **traceroute** não esteja funcionando (ou esteja operando de maneira incrivelmente lenta), pode ser que ele esteja passando por timeouts enquanto tenta determinar os nomes de hosts de gateways usando DNS. Se o DNS estiver falho no host por meio do qual está fazendo o rastreio, use **traceroute -n** para solicitar saída numérica. Essa opção impede o uso de DNS; pode ser que seja a única maneira de fazer com que **traceroute** funcione em uma rede defeituosa.

20.4 NETSTAT: OBTENHA TONELADAS DE ESTATÍSTICAS DE REDE

netstat fornece uma grande quantidade de informações sobre o estado do software de rede de seu computador, inclusive estatísticas de interface, informações de roteamento e tabelas de conexão. Realmente não há um tema unificador para os diferentes conjuntos de saída, exceto pelo fato de todos eles estarem relacionados com a rede.

Aqui, discutimos os quatro empregos mais comuns de **netstat**:

- Monitorar o estado das conexões de rede
- Inspecionar as informações de configuração de interface
- Examinar a tabela de roteamento
- Obter informações estatísticas operacionais para diversos protocolos de rede

Snow666

Monitorar o estado das conexões de rede

Sem usar nenhum argumento, **netstat** exibe o estado das portas TCP e UDP ativas. Servidores inativos ("em escuta") aguardando por conexões normalmente não são mostrados; eles podem ser visualizados via **netstat -a**. (Conexões para "sockets de domínio UNIX" também são mostradas, mas já que elas não estão relacionadas à rede, não as discutimos aqui.)

A saída ficará parecida com o seguinte:

```
% netstat -a
Conexões Internet ativas (servidores e estabelecidas)
Proto    Recv-Q    Send-Q    Endereço Local     Endereço Remoto    Estado
tcp        0         0       *.6013             *.*                OUÇA
tcp        0         0       *.6013             *.*                OUÇA
tcp        0         0       nimi.ssh           xor.com.4105       ESTABELECIDA
tcp        0         20      nimi.ssh           xor.com.1612       ESTABELECIDA
tcp        0         0       *.13500            *.*                OUÇA
tcp        0         0       nimi.ssh           135.197.2.114.883  ESTABELECIDA
tcp        0         0       nimi.1599          xor.com.telnet     ESTABELECIDA
tcp        0         0       *.ssh              *.*                OUÇA
tcp        0         0       *.ssh              *.*                OUÇA
tcp        0         0       nimi.ssh           135.197.2.114.776  ESTABELECIDA
tcp        0         0       *.cvsup            *.*                OUÇA
udp        0         0       *.syslog           *.*
udp        0         0       *.ntalk            *.*
...
```

Esse exemplo precedente foi executado no host nimi. Ele mostra várias conexões SSH que chegam, uma conexão **telnet** que sai e um bocado de portas escutando outras conexões.

Os endereços são mostrados no formato *nome-do-host.serviço*, onde *serviço* é o número de uma porta. Para serviços bem conhecidos, **netstat** mostra a porta simbolicamente, usando o mapeamento definido em **/etc/services**. Você pode obter endereços numéricos pela opção **-n**. Lembre-se: se seu DNS estiver interrompido, **netstat** será difícil de ser usado sem o flag **-n**.

`Send-Q` e `Recv-Q` ilustra os tamanhos das filas de envio e de recepção para a conexão no host local; os tamanhos de fila na outra extremidade de uma conexão TCP poderiam ser diferentes. Eles devem tender a 0 e, pelo menos, não ser consistentemente não-zero. Obviamente, se você estiver executando **netstat** por meio de um terminal de rede, a fila de envio para a sua conexão provavelmente jamais será 0.

O estado da conexão possui significado somente para TCP; UDP é um protocolo sem conexões. Os estados mais comuns que você verá são `ESTABELECIDA` para conexões ativas no momento, `OUÇA` para servidores aguardando conexões (normalmente não exibido sem a opção **-a**) e `TIME_WAIT` para conexões em processo de fechamento.

Essa tela é basicamente útil para depurar problemas de nível mais alto uma vez que você tenha determinado que os recursos básicos da rede estão funcionando corretamente. Ela permite que você verifique se os servidores estão configurados corretamente e facilita o diagnóstico de certos tipos de comunicações defeituosas, particularmente com TCP. Por exemplo, uma conexão que permanece no estado `SYN_ENVIADO` identifica um processo que está tentando contactar um servidor inexistente ou servidor de redes inacessível.

Veja o Capítulo 12 para obter maiores informações sobre ajustes de kernel.

Se **netstat** mostra um grande número de conexões na condição `SYN_WAIT`, seu host provavelmente está incapacitado de manipular o número de conexões que está sendo solicitado. Essa falta de adequação pode se dever às limitações de ajuste do kernel ou até mesmo a inundação maliciosa.

Inspecionando as informações de configuração de interface

netstat -i mostra o estado das interfaces de rede. Por exemplo, a seguir temos a saída gerada por **netstat -i** na máquina evolve:

```
% netstat -i
Tabela de Interfaces do Kernel
eth0        Encapsulamento do Link:Ethernet  Endereço de HW 00:02:B3:19:C8:82
            inet end.:192.168.2.1 Bcast:192.168.2.255 Mask:255.255.255.0
            UP BROADCAST RUNNING MULTICAST MTU:1500 Métrica:1
            RX packets:1121527 errors:0 dropped:0 overruns:0 frame:0
            TX packets:1138477 errors:0 dropped:0 overruns:0 carrier:0
```

Snow666


```
          colisões:0 txqueuelen:100
          Interrupt:7 Endereço de E/S:0xef00

eth1      Encapsulamento do Link:Ethernet  Endereço de HW 00:02:B3:19:C6:86
          inet end.:192.168.1.13 Bcast:192.168.1.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST MTU:1500 Métrica:1
          RX packets:67543 errors:0 dropped:0 overruns:0 frame:0
          TX packets:69652 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:100
          IRQ:5 Endereço de E/S:0xed00

lo        Encapsulamento do Link: Loopback Local
          inet end.:127.0.0.1 Mask:255.0.0.0
          UP LOOPBACK RUNNING MTU:3924 Métrica:1
          RX packets:310572 errors:0 dropped:0 overruns:0 frame:0
          TX packets:310572 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:0
```

Esse host possui duas interfaces de rede: uma para tráfego regular mais uma conexão "backlan". Pacotes `RX` e `TX` relatam o número de pacotes que foram recebidos e transmitidos em cada interface desde o momento em que a máquina foi inicializada. São contabilizados diversos tipos diferentes de erros nos depósitos de erros e é normal aparecerem alguns.

Os erros devem ser inferiores a 1% dos pacotes associados. Se sua taxa de erros for alta, compare as taxas de várias máquinas vizinhas. Um grande número de erros numa única máquina sugere problemas com a conexão ou com a interface daquela máquina. Uma taxa de erros alta em todos os pontos indica, mais provavelmente, problemas de mídia ou de rede.

Colisões indicam uma rede carregada; erros normalmente indicam problemas de cabeamento. Embora uma colisão seja um tipo de erro, ela é contabilizada separadamente por **netstat**. O campo chamado `Colisões` dá o número de colisões que ocorreram enquanto os pacotes estavam sendo enviados.[2] Use esse número para calcular a porcentagem de pacotes de saída (pacotes `TX`) que resultam em colisões. Numa rede funcionando de maneira apropriada, a taxa de colisões deve ser inferior a 3% do total de pacotes de saída, e algo um pouco acima de 10% indica sérios problemas de congestionamento.

Você também pode executar **netstat** com o flag **-c** para relatar continuamente informações estatísticas sobre as interfaces de rede. Este modo é especialmente útil para rastrear a fonte dos erros. **netstat -i** é capaz de alertá-lo sobre existência de problemas, mas ele não consegue lhe dizer se os erros provêm de um problema contínuo de baixo nível ou de um evento breve, porém catastrófico. Observar a rede ao longo do tempo sob uma série de condições de carga lhe dá uma visão muito melhor do que está acontecendo. Tente executar **ping** com um tamanho de pacote grande enquanto observa a saída de **netstat**.

Examinando a tabela de roteamento

netstat -r exibe a tabela de roteamento do kernel. Eis um exemplo de uma máquina com duas interfaces de rede:

```
% netstat -r -n
Tabela de roteamento IP do Kernel
Destino     Roteador        MáscaraGen.     Opções   MSS   Janela   irtt  Iface
192.168.1.0 0.0.0.0         255.255.255.0   U          0   0           0  eth1
192.168.2.0 0.0.0.0         255.255.255.0   U          0   0           0  eth0
127.0.0.0   0.0.0.0         255.0.0.0       U          0   0           0  lo
0.0.0.0     192.168.1.254   0.0.0.0         UG         0   0           0  eth1
```

Destinos e gateways (Roteador) podem ser exibidos tanto como nomes de host como endereços IP; o flag **-n** solicita saída numérica.

Consulte a página 260 para obter maiores informações sobre a tabela de roteamento.

`Flags(Opções)` caracterizam a rota: `U` significa up (ativo), `G` é um gateway e `H` é uma rota de host. O flag `D` (não exibido) indica a rota resultante de um redirecionamento ICMP. `G` e `H` juntos indicam uma rota de host que passa através de um gateway intermediário. Os campos restantes fornecem estatísticas sobre a rota: o número atual de conexões TCP usando a rota, o numero de pacotes enviados e a interface usada. Lembre-se de que essa saída varia ligeiramente entre os diversos sistemas operacionais.

2. Esse campo possui significado somente em redes baseadas em CSMA/CD como Ethernet.

www.snow666.blogspot.com
Snow666

Use essa forma de **netstat** para verificar a saúde da tabela de roteamento de sua máquina. É particularmente importante verificar se o sistema possui uma rota-padrão e que esta rota seja correta. A rota-padrão é representada por um endereço de destino todo preenchido com 0s (0.0.0.0).

Visualizando estatísticas operacionais para vários protocolos de rede

netstat -s faz um dump do conteúdo dos contadores que são espalhados através do código de rede. A saída possui seções separadas para IP, ICMP, TCP e UDP. A seguir apresentamos trechos da saída gerada por **netstat -s** de uma máquina gateway; eles foram editados para mostrar somente os trechos de informação mais interessantes.

```
ip:
      1396128 total packets received
      0 forwarded
      0 incoming packets discarded
      139221 incoming packets delivered
      1422181 requests sent out

icmp:
      7917 ICMP messages received
      30 input ICMP message failed.

ICMP input histogram:
      destination unreachable: 3093
      timeout in transit: 64
      source quenchs: 2
      echo requests: 4704
4848 ICMP messages sent
0 ICMP messages failed
ICMP output histogram:
      destination unreachable: 144
      echo replies: 4704
```

Os números de solicitações de eco, respostas geradas e respostas de eco são todos coincidentes. Note que mensagens "destination unreachable" (destino inatingível) ainda podem ser geradas mesmo quando todos os pacotes são aparentemente encaminháveis. Pacotes ruins podem eventualmente atingir um gateway que os rejeita, e mensagens de erros são então enviadas de volta através da cadeia de gateway.

```
tcp:
      10753 active connections openings
      0 passive connection openings
      106 failed connection attempts
      0 connection resets received
      4 connections established
      1248990 segments received
      1276311 segments send out
      35946 segments retransmited
      13 bad segments received.
      3080 resets sent
```

É uma boa idéia desenvolver um entendimento sobre os intervalos normais dessas estatísticas para que você possa reconhecer estados patológicos.

20.5 Analisadores de Pacotes

tcpdump e Ethereal pertencem à mesma classe de ferramentas conhecidas como analisadores ("sniffers") de pacotes. Elas escutam o tráfego numa rede e registram ou imprimem pacotes que atendam a certos critérios especificados pelo usuário. Por exemplo, todos os pacotes enviados de ou para um determinado host ou pacotes TCP relacionados a uma determinada conexão de rede poderiam ser inspecionados.

Analisadores de pacotes são úteis tanto na solução de problemas conhecidos como na descoberta de problemas inteiramente novos. É uma boa idéia fazer uma análise ocasional de sua rede para certificar-se de que o tráfego está em ordem.

Já que analisadores de pacotes precisam estar aptos a interceptar tráfego que a máquina local não receberia normalmente (ou pelo menos, prestaria atenção a), o hardware da rede subjacente tem de permitir acesso a todos os

Snow666

pacotes. As tecnologias de difusão ampla como Ethernet funcionam bem, assim como alguns tipos de redes token ring nas quais o remetente de um pacote o remove do anel após ele ter feito um circuito completo.

Consulte a página 255 para obter maiores informações sobre comutadores (switches) de rede.

Como os analisadores de pacotes precisam ver a maior quantidade possível de tráfego de rede bruto, eles podem ser impedidos por comutadores de rede que, por projeto, tentam limitar a propagação de pacotes "desnecessários". Entretanto, ainda pode ser informativo experimentar um analisador numa rede com comutadores. Talvez você descubra problemas relacionados com pacotes de difusão ampla ou multidifusão. Dependendo do fabricante de seu comutador, talvez você possa ser surpreendido com o nível de tráfego que poderá ver.

Além de ter acesso potencial a todos os pacotes de rede, o hardware da interface tem de fornecer uma maneira de realmente transportar esses pacotes para a camada de software. Os endereços de pacotes são normalmente verificados em hardware e somente pacotes de difusão ampla/multidifusão e aqueles endereçados ao host local são retransmitidos ao kernel. Em "modo promíscuo" uma interface deixa que o kernel leia todos os pacotes da rede, mesmo aqueles destinados a outros hosts.

Os analisadores de pacotes entendem vários dos formatos de pacotes usados por daemons-padrão e normalmente são capazes de colocar esses pacotes num formato legível para o usuário. Essa capacidade torna mais fácil rastrear o fluxo de uma conversação entre dois programas. Alguns analisadores imprimem o conteúdo ASCII de um pacote além do cabeçalho do pacote, que pode ser útil para investigar protocolos de camadas mais altas. Já que alguns desses protocolos enviam informações (e até mesmo senhas) através da rede como texto puro, você tem de tomar cuidado para evitar invadir a privacidade de seus usuários.

Analisadores têm de ler dados de um dispositivo de rede bruto, assim eles têm de ser executados como root. Apesar dessa limitação servir para diminuir a chance de que usuários comuns possam escutar o tráfego de sua rede, ela realmente não é uma grande barreira. Alguns sites optam por eliminar programas analisadores da maioria dos hosts para reduzir a chance de abuso. Se não houver mais nada, você deve verificar as interfaces de seu sistema para ter certeza de que elas não estão sendo executadas em modo promíscuo sem o seu conhecimento ou consentimento. Em sistemas Linux, uma interface em modo promíscuo exibirá o flag `PROMISC` em sua saída de estado **ifconfig**. Você também pode verificar sua rede em busca de interfaces executadas em modo promiscuo através de ferramentas como o PromiScan (disponível em www.securityfriday.com).

tcpdump: o rei dos analisadores

tcpdump, mais uma incrível ferramenta de rede criada por Van Jacobson, é distribuída com a maioria das distribuições Linux. Ele vem sendo o padrão de indústria dos analisadores há muito tempo; a maioria das demais ferramentas de análise de rede lê e/ou rastreia arquivos em "formato tcpdump".

Como padrão, **tcpdump** escuta a primeira interface de rede com a qual ele se depara. Se ele optar pela interface errada, você poderá forçar uma dada interface por meio do flag **-i**. Se o DNS estiver interrompido ou você simplesmente não quiser que **tcpdump** realize pesquisas de nomes, use a opção **-n**. Essa opção é importante, pois serviço DNS lento pode fazer com que o filtro comece a perder pacotes antes de poderem ser tratados por **tcpdump**. O flag **-v** aumenta o nível de informações que pode ser visto sobre os pacotes e **-vv** lhe fornece ainda mais dados. Finalmente, **tcpdump** pode armazenar pacotes num arquivo com o flag **-w** e pode lê-los de volta com o flag **-r**.

Por exemplo, a saída a seguir provém da máquina jaguar.xor.com. A especificação de filtros **host jaguar** limita a exibição de pacotes àqueles que envolvem diretamente a máquina jaguar, seja como fonte seja como destino.

```
# tcpdump host jaguar
13:40:23 jaguar.xor.com.1697 > xor.com.domain: A? cs.colorado.edu.
13:40:23 xor.com.domain > jaguar.xor.com.1697: A mroe.cs.colorado.edu
13:40:23 jaguar.xor.com.1698 > xor.com.domain: PTR? 5.96.138.128.in-addr.arpa.
13:40:23 xor.com.domain > jaguar.xor.com.1698: PTR mroe.cs.colorado.edu.
```

O primeiro pacote mostra a máquina jaguar enviando uma solicitação DNS sobre cs.colorado.edu para xor.com. A resposta é o verdadeiro nome da máquina para o qual essa máquina é um nome alternativo, que é mroe.cs.colorado.edu. O terceiro pacote é uma pesquisa reversa do endereço IP de mroe e o quarto pacote contém a resposta esperada.

A página de manual de **tcpdump** contém diversos exemplos interessantes sobre filtragem avançada juntamente com uma lista completa das primitivas.

Ethereal: analisador visual

Se você for mais propenso a usar um programa aponte-e-clique para análise de pacotes, então Ethereal será o mais indicado. Disponível sob a GNU General Public License em www.ethereal.com, o Ethereal é um analisador de pa-

Snow666

cotes com interface gráfica com o usuário baseada no GTK+(kit de ferramentas GIMP) que possui maior funcionalidade do que a maioria dos produtos comerciais para análise. Ele pode ser executado em seu desktop Linux ou, se o seu laptop ainda estiver sofrendo terrivelmente nas sombrias eras do Windows, você também poderá fazer o download de seus binários.

Além dos pacotes para análise, o Ethereal possui uma série de recursos que o tornam muito prático. Um excelente recurso é sua capacidade de ler e gravar um grande número de outros formatos de arquivos de rastreio de pacotes, incluindo (porém não limitado somente a) os seguintes:

- tcpdump
- Sniffer da NAI
- Sniffer Pro
- NetXray
- **snoop**
- Shomiti Surveyor
- Network Monitor da Microsoft
- LANalyzer da Novell
- Cisco Secure IDS iplog

O segundo recurso superútil é você poder clicar sobre um pacote num stream TCP e solicitar ao Ethereal para "remontar" (juntar) os dados da carga útil de todos os pacotes contidos no fluxo (stream). Esse recurso é útil caso você queira examinar rapidamente os dados transferidos durante uma conversação TCP completa, como uma conexão transportando uma mensagem de e-mail através da rede.[3]

20.6 Protocolos de Gerenciamento de Redes

As redes cresceram rapidamente, tanto em termos de tamanho quanto em valor, ao longo da última década e juntamente com este crescimento veio a necessidade de uma maneira eficiente de gerenciá-las. Fabricantes e organizações de padrões abordaram esse desafio de várias maneiras diferentes. Os desenvolvimentos mais significativos foram a introdução de vários protocolos de gerenciamento de dispositivos padrão e uma fartura de produtos de alto nível que exploram estes protocolos.

Os protocolos para gerenciamento de redes fornecem uma maneira-padrão de sondar um dispositivo para descobrir sua configuração, estado e conexões de rede. Além disso, eles permitem que algumas dessas informações possam ser modificadas para que o gerenciamento de rede possa ser padronizado através de diversos tipos de máquinas e realizado em uma localização centralizada.

O protocolo para gerenciamento mais comumente usado com o TCP/IP é o SNMP (Simple Network Management Protocol, protocolo simples de administração de rede). Apesar de seu nome, o SNMP é, na verdade, bem complexo. Ele define um espaço de nomes hierárquico de dados de administração e uma maneira de ler e gravar os dados em cada um dos nós. Ele também define um modo para entidades gerenciadas ("agentes") enviarem mensagens de notificação de eventos ("capturas") para estações de gerenciamento. O protocolo em si é simples; a maior parte da complexidade do SNMP reside acima da camada do protocolo nas convenções para construção do espaço de nomes e nas convenções para formatar itens de dados dentro de um nó. O SNMP é suportado amplamente.

Vários outros padrões estão pairando por aí. Muitos deles se originaram da DMTF (Distributed Management Task Force, força de trabalho de administração distribuída), que é responsável por conceitos como o WBEM (Web-Based Enterprise Management, administração de empresa baseada em Web), o DMI (Desktop Management Interface, interface de administração de desktop) e o CIM (Conceptual Interface Model, modelo conceitual de interface). Alguns desses conceitos, particularmente o DMI, foram abarcados por vários fornecedores importantes e podem se tornar um complemento útil (ou até mesmo um substitutivo) para o SNMP. Por enquanto, porém, a vasta maioria do gerenciamento de rede ocorre sobre SNMP.

Já que o SNMP é somente um protocolo abstrato, você precisa tanto de um programa servidor ("agente") e um cliente ("gerente") para fazer uso dele. (Talvez de forma contrária à intuição, o lado servidor do SNMP representa aquilo que está sendo gerenciado, e o lado cliente é o gerente.) Clientes variam desde utilitários de linha de comando simples a estações de gerenciamento dedicadas que exibem graficamente redes e falhas em cores que saltam aos olhos.

3. Você pode usar o utilitário **tcpflow** para realizar um recurso similar na linha de comando em um rastreio **tcpdump**.

Snow666

Estações dedicadas para gerenciamento de redes é a principal razão da existência dos protocolos de gerenciamento. A maioria dos produtos permite que você crie um modelo topográfico da rede, bem como um modelo lógico; os dois são apresentados juntos na tela, juntamente com uma indicação contínua do estado de cada componente.

Da mesma maneira que um gráfico é capaz de revelar o significado oculto de uma página cheia de números, uma estação para gerenciamento de redes é capaz de sintetizar o estado de uma grande rede de um modo facilmente aceito pelo cérebro humano. Este tipo de sumário executivo é quase impossível de se obter de uma outra maneira.

Uma grande vantagem do gerenciamento por protocolo é o fato de ele colocar todos os tipos de hardware de rede num campo de nível de atuação. Os sistemas Linux são, basicamente, todos similares, mas roteadores, switches e outros componentes de baixo nível não são. Com SNMP, todos falam a mesma língua e podem ser sondados, reinicializados e configurados de um ponto central. É bom ter uma interface consistente com todo o hardware da rede.

20.7 SNMP: Simple Network Management Protocol

Quando o SNMP começou a ser usado largamente no início dos anos 90, ele começou uma minicorrida do ouro. Centenas de empresas lançaram pacotes de gerenciamento SNMP. Da mesma forma, muitos fabricantes de hardware e de software começam a colocar um agente SNMP como parte de seus produtos.

Antes de mergulharmos nos detalhes do SNMP, devemos notar que a terminologia associada a ele é um dos jargões mais deploráveis encontrados no campo das redes computacionais. Os nomes-padrão para conceitos e objetos do SNMP o induzirão a se afastar de um entendimento do que está acontecendo. Os responsáveis por esse estado de coisas deviam ter seus teclados destruídos.

Organização SNMP

Os dados SNMP são organizados segundo uma hierarquia padronizada. Essa organização forçada permite que o espaço de dados permaneça universal e extensível ao mesmo tempo, pelo menos teoricamente. Grandes porções são reservadas para expansões futuras e adições específicas de fornecedor são localizadas para evitar conflitos. A hierarquia de atribuição de nomes é formada por MIBs ("Management Information Bases", bases de informação de gerenciamento), arquivos de texto estruturados que descrevem os dados que podem ser acessados via SNMP. Os MIBs contêm descrições de variáveis de dados específicas, referidas através de nomes conhecidos como OIDs (Object Identifiers, identificadores de objetos).

Traduzindo para o português, isso significa que o SNMP define um espaço de nomes hierárquico de variáveis cujos valores estão amarrados a parâmetros "interessantes" do sistema.

Os tipos de dados básicos que uma variável SNMP pode conter são inteiro, string e nulo. Estes podem ser combinados em seqüências de tipos básicos, e uma seqüência pode ser instanciada repetidamente para formar uma tabela. A maioria das implementações também suporta uma série de outros tipos de dados.

A hierarquia SNMP é muito parecida com a de um sistema de arquivos. Porém, é usado um ponto como caractere separador e cada nó recebe um número em vez de um nome. Por convenção, os nós também recebem nomes textuais para facilidade de referência, porém esta atribuição de nomes é, na realidade, apenas uma conveniência de alto nível, e não um recurso da hierarquia (ela é similar, em princípio, à associação de nomes de hosts a endereços IP).

Por exemplo, o OID que se refere ao tempo em que o sistema fica ativo é 1.3.6.1.2.1.1.3. Esse OID também é conhecido pelo nome legível iso.org.dod.internet.mgmt.mib-2.system.sysUpTime.

Os níveis altos da hierarquia SNMP são artefatos políticos e geralmente não contêm dados úteis. De fato, dados úteis podem ser encontrados atualmente somente abaixo do OID iso.org.dod.internet.mgmt (numericamente, 1.3.6.1.2).

O MIB SNMP básico para TCP/IP (MIB-I) define o acesso a dados de gerenciamento comuns: informações sobre o sistema, suas interfaces, conversão de endereços e operações de protocolos (IP, ICMP, TCP, UDP, e outros). Um trabalho refeito posteriormente e mais completo desse MIB (denominado MIB-II) é definido na RFC1213. A maioria dos fabricantes que fornecem um servidor SNMP suporta o MIB-II. A Tabela 20.1 apresenta uma amostra de nós do espaço de nomes MIB-II.

Tabela 20.1 OIDs selecionados de MIB-II.

OID[a]	Tipo	Conteúdo
system.sysDescr	string	Informações de sistema: fabricante, modelo, tipo de sistema operacional etc.
system.sysLocation	string	Localização física da máquina
system.sysContact	string	Informações para contato para o proprietário da máquina

www.snow666.blogspot.com
Snow666

Tabela 20.1 OIDs selecionados de MIB-II. (*continuação*)

OID[a]	Tipo	Conteúdo
system.sysName	string	Nome do sistema, normalmente o nome DNS completo
interfaces.ifNumber	int	Número de interfaces de rede presentes
interfaces.ifTable	table	Tabela de informações sobre cada interface
ip.ipForwarding	int	1 se o sistema for um gateway; caso contrário, 2
ip.ipAddrTable	table	Tabela de dados de endereçamento IP (máscaras etc.)
ip.ipRouteTable	table	A tabela de roteamento do sistema
icmp.icmpInRedirects	int	Número de redirecionamentos ICMP recebidos
icmp.icmpInEchos	int	Número de pings recebidos
tcp.tcpConnTable	table	Tabela de conexões TCP atuais
udp.udpTable	table	Tabela de sockets UDP com servidores escutando

a. Relativo a iso.org.dod.internet.mgmt.mib-2.

Além do MIB básico, há MIBs para vários tipos de protocolos e interfaces de hardware. Há MIBs para fabricantes específicos e MIBs para produtos de hardware particulares. Um MIB para você, um MIB para mim, e muitos outros.

Um MIB é somente uma convenção em relação à atribuição de nomes de dados para gerenciamento. Para ser útil, um MIB tem de ser emparelhado com o código executado no agente que faz a associação entre o espaço de nomes SNMP e o real estado do dispositivo. O agente Linux padrão já vem com código para o MIB básico (agora MIB-II). Alguns agentes são extensíveis para poder incluir MIBs suplementares, e outros não o são.

Operações de protocolos SNMP

Há somente quatro operações SNMP básicas: get, get-next, set e trap.

Get e set são as operações básicas para leitura e gravação de dados num nó identificado por um OID específico. Get-next é usado para avançar através de uma hierarquia MIB, bem como ler o conteúdo de tabelas.

Trap é uma notificação assíncrona e não solicitada do servidor (agente) ao cliente (gerente) que relata a ocorrência de um evento ou condição interessante. São definidos vários traps-padrão, inclusive notificações "I've just come up" (acabei de ser iniciado), traps que relatam a falha ou a recuperação de um link de rede e traps para vários problemas de roteamento e de autenticação. Muitos outros traps não tão padronizados estão sendo usados comumente, inclusive alguns que apenas observam os valores de outras variáveis SNMP e disparam uma mensagem quando um intervalo especificado é excedido. O mecanismo pelo qual os destinos de mensagens trap são especificados depende da implementação do agente.

Já que mensagens SNMP têm um grande potencial para modificar informações de configuração, é necessário algum mecanismo de segurança. A versão mais simples de segurança SNMP se baseia no conceito de um "nome comunitário" SNMP que na verdade é apenas uma maneira horrivelmente confusa de dizer "senha". Normalmente há um nome comunitário para acesso somente de leitura e um outro que permite gravação.

A versão 3 do padrão SNMP introduziu métodos de controle de acesso com maior segurança. Embora o suporte a esses esquemas ainda seja um tanto limitado em hardware de rede de produção, é esperado que esta situação mude em breve.

RMON: MIB para monitoramento remoto

O MIB RMON permite a coleta de dados genéricos de desempenho da rede (isto é, dados não amarrados a nenhum dispositivo em particular). Os analisadores de rede ou "sondas" podem ser empregados pela rede para coletar informações sobre utilização e desempenho. Assim que uma quantidade de dados razoável tiver sido coletada, estatísticas e informações interessantes sobre os dados poderão ser devolvidas para uma estação de gerenciamento central para análise e apresentação. Muitas sondas possuem um buffer de captura de pacotes e são capazes de fornecer uma espécie de recurso **tcpdump** remoto.

RMON é definido na RFC1757, que se tornou um padrão preliminar em 1995. O MIB se divide em nove "grupos RMON". Cada grupo contém um conjunto diferente de estatísticas de rede. Se você tiver uma rede grande com muitas conexões WAN, deve considerar a possibilidade de adquirir sondas para reduzir o tráfego SNMP através de seus links WAN. Assim que tiver acesso a sumários estatísticos das sondas RMON, normalmente não haverá mais

www.snow666.blogspot.com
Snow666

necessidade de coletar dados brutos remotamente. Muitos comutadores e roteadores suportam o RMON e armazenarão pelo menos algumas estatísticas da rede.

20.8 O AGENTE NET-SMNP

Quando o agente SNMP foi padronizado pela primeira vez, tanto a Carnegie Mellon University quanto o MIT produziram suas implementações. A implementação da CMU era mais completa e rapidamente se tornou um padrão de fato. Quando o desenvolvimento ativo na CMU caiu bastante, os pesquisadores da UC Davis assumiram o software. Após estabilizarem o código, eles transferiram a manutenção em andamento para o repositório SourceForge. O pacote é agora conhecido como NET-SNMP.

A distribuição NET-SNMP agora é a implementação SNMP gratuita autorizada para o Linux. Ela inclui um agente SNMP, algumas ferramentas em linha de comando e até mesmo uma biblioteca para desenvolvimento de aplicações que entendem SNMP. Discutimos o agente com um certo nível de detalhes aqui e na próxima página daremos uma olhada nas ferramentas de linha de comando. A versão mais recente se encontra disponível na Web em net-snmp.sourceforge.net.

Como ocorre em outras implementações, o agente coleta informações sobre o host local e o serve aos gerentes SNMP através da rede. A instalação padrão inclui MIBs para estatísticas de interfaces de rede, memória, disco, processos e CPU. O agente pode ser facilmente ampliado já que ele é capaz de executar um comando Linux arbitrário e retornar a saída gerada pelo comando na forma de uma resposta SNMP. Você pode usar esse recurso para monitorar praticamente qualquer coisa em seu sistema com SNMP.

Por padrão, o agente é instalado como **/usr/sbin/snmpd**. Normalmente ele é iniciado no momento da inicialização e extrai informações relativas à configuração de arquivos contidos no diretório **/etc/snmp**. O mais importante desses arquivos é **snmpd.conf**, que contém a maior parte das informações de configuração e vem com uma série de modelos para métodos de coleta de dados já habilitados. Embora a intenção dos autores do NET-SNMP parece ter sido a dos usuários poderem editar somente o arquivo **snmpd.local.conf**, é necessário editar o arquivo **snmpd.conf** pelo menos uma vez para desabilitar qualquer método de coleta de dados padrão que você não pretenda usar.

O script **configure** do NET-SNMP permite especificar um arquivo de log padrão e uma série de outras configurações locais. Podemos usr **snmpd -l** para especificar um arquivo de log alternativo ou **-s** para direcionar mensagens de log para syslog. A Tabela 20.2 apresenta uma lista dos flags mais importantes de **snmpd**. Recomendamos que você sempre use o flag **-a**. Para depuração, você deve usr os flags **-V**, **-d** ou **-D**, cada um dos quais fornecendo progressivamente mais informações.

Tabela 20.2 Flags úteis para o snmpd do NET-SNMP.

Flag	Função
-l *logfile*	Registra informações em logfile
-a	Registra em log os endereços de todas as conexões SNMP
-d	Registra em log o conteúdo de cada pacote SNMP
-V	Habilita o registro em log prolixo
-D	Registra em log informações de depuração (um bocado delas)
-h	Exibe todos os argumentos para **snmpd**
-H	Exibe todas as diretivas de arquivos de configuração
-A	Anexa informações ao arquivo de log em vez de escrever por cima dele
-s	Registra em log informações no syslog (usa o recurso daemon)

Vale a pena mencionar que há disponíveis muitos módulos Perl úteis relacionados com SNMP. Vá ao CPAN[4] para obter as informações mais recentes caso esteja interessado em escrever os seus próprios scripts para gerenciamento de redes.

20.9 APLICATIVOS PARA GERENCIAMENTO DE REDES

Começamos esta seção explorando as ferramentas de gerenciamento SNMP mais simples: os comandos fornecidos com o pacote NET-SNMP. Esses comandos são úteis para familiarização com SNMP e também são excelentes para verificações únicas de OIDs específicos. A seguir, damos uma olhada em MRTG, um programa que gera gráficos

4. CPAN (Comprehensive Perl Archive Network) é um conjunto incrível de módulos Perl úteis. Veja em www.cpan.org.

Snow666

com históricos de valores SNMP, e NOCOL, um sistema de monitoramento baseado em eventos. Concluímos com algumas recomendações do que analisar ao comprar um sistema comercial.

As ferramentas NET-SNMP

Mesmo que seu sistema venha com um servidor SNMP próprio, você ainda pode querer compilar e instalar as ferramentas do lado cliente, listadas na Tabela 20.3, do pacote NET-SNMP.

Tabela 20.3 Ferramentas de linha de comando no pacote NET-SNMP.

Comando	Função
snmpdelta	Monitora, ao longo do tempo, alterações em variáveis SNMP
snmpdf	Monitora espaço em disco num host remoto via SNMP
snmpget	Obtém o valor de uma variável SNMP por meio de um agente
snmpgetnext	Obtém a próxima variável na seqüência
snmpset	Configura uma variável SNMP num agente
snmptable	Obtém uma tabela das variáveis SNMP
snmptranslate	Pesquisa e descreve OIDs na hierarquia MIB
snmptrap	Gera um alerta de captura
snmpwalk	Percorre um MIB iniciando num dado OID

Além dos valores na linha de comandos, esses programas são tremendamente práticos em scripts simples. Normalmente é útil fazer com que **snmpget** salve valores de dados interessantes num arquivo de texto a cada minuto. (Use **cron** para implementar o cronograma; veja o Capítulo 9, *Processos Periódicos*.)

snmpwalk é mais uma ferramenta útil. Iniciando num OID especificado (ou no início do MIB, por padrão), esse comando realiza repetidamente chamadas "get next" a um agente. Este comportamento resulta numa lista completa de OIDs disponíveis e seus valores associados. Eis um exemplo de **snmpwalk** do host jaguar ("public" é a string comunitária):

```
% snmpwalk jaguar public
system.sysDescr.0 = Linux jaguar 2.2.12-20 #1 Mon Sep 27 10:40:35 EDT 1999
system.sysUpTime.0 = Timeticks: (88516617) 10 days, 5:52:46.17
system.sysName.0 = jaguar
system.sysLocation.0 = Second Floor Machine Room
interfaces.ifNumber.0 = 2
interfaces.ifTable.ifEntry.ifIndex.1 = 1
interfaces.ifTable.ifEntry.ifIndex.2 = 2
interfaces.ifTable.ifEntry.ifDescr.1 = "lo0" Hex: 6C 6F 30
interfaces.ifTable.ifEntry.ifDescr.2 = "eth0" Hex: 65 74 68 30
interfaces.ifTable.ifEntry.ifType.1 = softwareLoopback(24)
interfaces.ifTable.ifEntry.ifType.2 = ethernet-csmacd(6)
interfaces.ifTable.ifEntry.ifMtu.1 = 3924
interfaces.ifTable.ifEntry.ifMtu.2 = 1500
interfaces.ifTable.ifEntry.ifInOctets.1 = 12590602
interfaces.ifTable.ifEntry.ifInOctets.2 = 2287718531
interfaces.ifTable.ifEntry.ifInErrors.1 = 0
interfaces.ifTable.ifEntry.ifInErrors.2 = 218
Interfaces.ifTable.ifEntry.ifOutOctets.1 = 12591593
Interfaces.ifTable.ifEntry.ifOutOctets.2 = 3374588125
...
```

Nesse exemplo, vemos algumas informações genéricas sobre o sistema, seguidas por estatísticas sobre as interfaces de rede do host, lo0 e eth0. Dependendo de os MIBs suportados pelo agente que você está gerenciando, um dump completo pode chegar a centenas de linhas.

www.snow666.blogspot.com
Snow666

MRTG: Multi-Router Traffic Grapher

MRTG, criado por Tobi Oetiker da ETH em Zurique, coleta dados SNMP ao longo do tempo e depois os coloca em gráfico. Ele é escrito em sua maior parte em Perl. O MRTG é valiosíssimo para analisar o uso histórico dos sistemas e dos recursos de rede.

O MRTG é executado regularmente do **cron** e pode coletar dados de qualquer fonte SNMP. Cada vez em que o programa é executado, novos dados são armazenados e imagens, criadas.

O MRTG é gratuito e oferece diversos recursos atraentes. Primeiramente, ele possui um banco de dados dimensionado estaticamente e de manutenção zero; o software armazena somente dados suficientes para criar os gráficos necessários. Por exemplo, MRTG poderia armazenar uma amostra a cada minuto durante um dia, uma amostra a cada hora durante uma semana e uma amostra por semana durante um ano. Esse esquema de consolidação permite que você preserve informações históricas importantes sem ter de armazenar detalhes não importantes ou perder tempo com a administração do banco de dados.

Em segundo lugar, MRTG pode registrar e colocar em forma de gráfico qualquer variável SNMP. Você fica livre para coletar quaisquer dados que quiser. Quando combinado com o agente NET-SNMP, o MRTG pode fornecer uma perspectiva histórica de praticamente qualquer sistema ou recurso de rede.

O Diagrama A mostra alguns exemplos dos gráficos gerados por MRTG. Estes gráficos mostram o tráfego numa interface de rede ao longo de períodos de um dia e uma semana.

O futuro do MRTG está num novo pacote, RRDtool, do mesmo autor. RRDtool é conceitualmente similar ao MRTG, porém com consolidação de dados e recursos de gráficos melhores. Diferentemente do MRTG, o RRDtool não oferece quaisquer métodos de coleta de dados próprios. Em vez disso, um software distinto tem de coletar os dados.

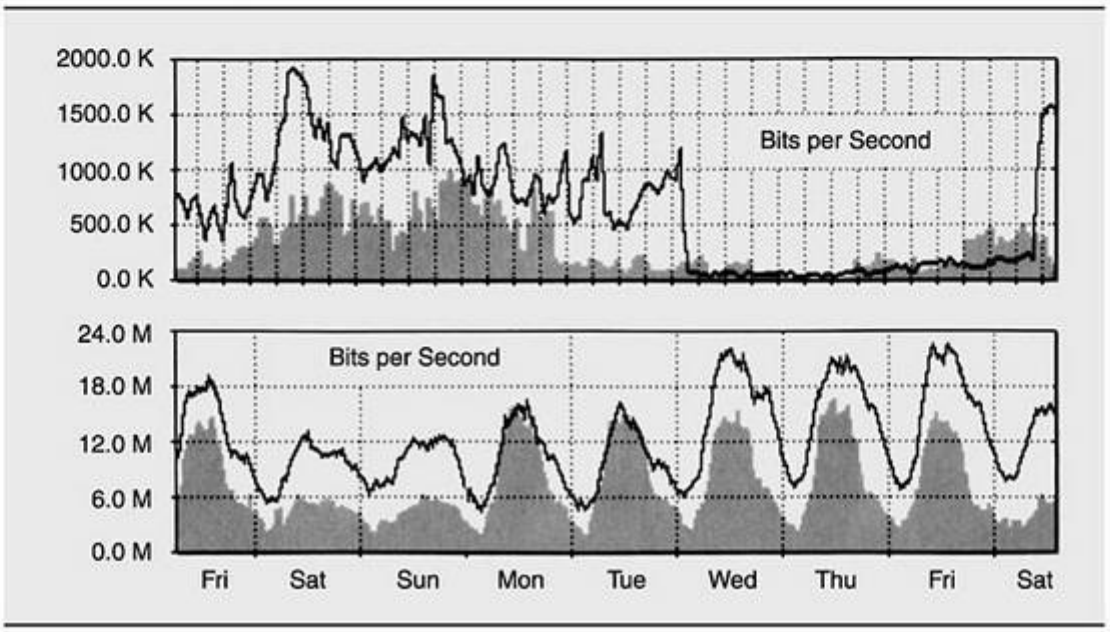
Atualmente, a ferramentas Cricket de Jeff Allen é a melhor opção para essa função. Cricket não se limita a coletar dados SNMP; ele é capaz de extrair dados de praticamente qualquer fonte de rede. Já que ele é escrito em Perl, você poderá facilmente adicionar novas fontes de dados.

A homepage de Tobi Oetiker em ee-staff.ethz.ch/~oetiker fornece links para as versões atuais do MRTG, do RRDtool e do Cricket.

NOCOL: Network Operation Center On-Line

NOCOL é uma ferramenta para gerenciamento dirigida a eventos que é mantida atualmente por Vikas Aggarwal. Embora não vá ajudá-lo a determinar qual foi o aumento da utilização de largura de banda no último mês, ela emitirá uma mensagem via pager quando o seu servidor Web cair. Na realidade, NOCOL pode ser configurada para enviar uma mensagem via pager (ou e-mail) para o seu pessoal de operações após vários tipos de eventos.

Diagrama A Exemplos de gráficos gerados por MRTG



Snow666

A distribuição inclui programas de monitoramento que supervisionam uma série de pontos de falhas comuns. Você pode disparar novos monitoramentos em Perl ou até mesmo em C caso esteja se sentindo seguro. Para métodos de notificação, a distribuição pode enviar e-mail, gerar relatórios Web, visualizar o estado com uma interface **curses** e usar um modem discado para lhe remeter mensagens via pager. Como acontece com programas de monitoramento, é fácil fazer progressos por conta própria.

Se você não puder comprar uma ferramenta para gerenciamento de rede comercial, sugerimos considerar em especial o NOCOL. O software funciona muito bem para redes com menos de 100 hosts e dispositivos. Você pode ler mais a respeito em www.netplex-tech.com.

Plataformas comerciais para gerenciamento

Centenas de empresas vendem software para gerenciamento de redes, e novos competidores entram no mercado toda semana. Em vez de recomendar os produtos mais quentes do momento (que talvez podem não mais existir no instante em que este livro for impresso), tentaremos identificar os recursos que você deve procurar num sistema para gerenciamento de redes.

Flexibilidade de coleta de dados: É importante para as ferramentas de gerenciamento serem capazes de coletar dados de outras fontes diferentes do SNMP. Muitos pacotes incluem a habilidade de coletar dados de praticamente qualquer serviço de rede. Por exemplo, alguns pacotes podem realizar consultas a bancos de dados SQL, verificar registros DNS e conectar a servidores Web.

Qualidade da interface com o usuário: Sistemas caros normalmente oferecem uma interface gráfica com o usuário personalizada ou uma interface Web. A maioria dos pacotes bem comercializados hoje promove a capacidade de entender gabaritos XML para apresentação de dados. Embora a interface com o usuário normalmente se pareça mais com supervalorização de marketing, é importante ter uma interface que retransmita informações de forma clara, simples e compreensível.

Valor: Alguns pacotes para gerenciamento têm um preço salgado. O OpenView da HP é um dos mais caros e ao mesmo tempo um dos sistemas para gerenciamento de redes mais amplamente adotados para muitas corporações, há um valor inquestionável em poder dizer que seus sites são gerenciados por um sistema comercial topo de linha. Se isso não for importante para a sua empresa, você deve olhar na outra ponta do espectro para ferramentas gratuitas como MRTG e NOCOL.

Descoberta automatizada: Muitos sistemas oferecem a capacidade de "descobrir" sua rede. Por meio de uma combinação de pings de difusão ampla, solicitações SNMP, pesquisas de tabelas ARP e consultas DNS, eles são capazes de identificar todos os seus hosts e dispositivos locais. Todas as implementações de descoberta que vimos funcionam muito bem, mas nenhuma delas é muito precisa em uma rede complexa (ou protegida de forma muito intensa por firewalls).

Recursos de relatórios: Muitos produtos podem enviar e-mails de alerta, ativar pagers e gerar automaticamente tíquetes para sistemas de rastreio de problemas populares. Certifique-se de que a plataforma escolhida permita relatórios flexíveis; quem pode dizer com que tipos de dispositivos eletrônicos você estará lidando daqui a alguns anos?

Gerenciamento de configurações: Alguns fabricantes vão muito além do monitoramento e do alerta. Eles oferecem a capacidade de gerenciar o host em si e as configurações de dispositivos. Por exemplo, o CiscoWorks fornece uma interface que nos permite alterar a configuração de um roteador além de monitorar o seu estado com o SNMP. Pelo fato de as informações de configuração de dispositivo abrirem espaço para uma análise mais profunda dos problemas de rede, prevemos que serão desenvolvidos muitos pacotes segundo estas linhas no futuro.

20.10 Leitura Recomendada

CISCO ONLINE. *Internetworking Technology Overview: SNMP.*
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm.
HUNT, CRAIG; GIGI ESTABROOK. *TCP/IP Network Administration. 2. ed.* Sebastopol: O'Reilly & Associates. 1998.
STALLINGS, WILLIAM. Snmp, Snmpv2, Snmpv3, and Rmon 1 and 2. 3. ed. Reading, MA: Addison-Wesley. 1999.

Talvez você também possa achar úteis as seguintes RFCs. Em vez de citarmos os verdadeiros títulos das RFCs, descrevemos seu conteúdo. Os verdadeiros títulos são uma mistura de termos técnicos da moda e jargões SNMP.

- RFC1155 – Características do espaço de dados SNMP (tipos de dados etc.)
- RFC1156 – Definições MIB-I (descrição dos verdadeiros OIDs)
- RFC1157 – Simple Network Management Protocol (protocolo simples de administração de rede)
- RFC1213 – Definições MIB-II (OIDs)

Snow666

- RFCs 1901-1910 – SNMPv2
- RFC2011 – SNMPv2 MIB para IP
- RFC2012 – SNMPv2 MIB para TCP
- RFC2013 – SNMPv2 MIB para UDP
- RFC2021 – RMON Versão 2 usando SMIv2
- RFC2570 – Introdução ao SNMPv3

20.11 EXERCÍCIOS

E20.1 Você está diagnosticando um problema de rede, e **netstat -rn** gera a seguinte saída. Qual é o problema e que comando você usaria para corrigi-lo?

```
Destino          Roteador        MáscaraGen.     Opções  MSS  Janela     irtt    Iface
128.138.202.0    0.0.0.0         255.255.255.0   U        40  0            0     eth0
127.0.0.0        0.0.0.0         255.0.0.0       U        40  0            0     lo
```

★ **E20.2** Crie um script que monitore um dado conjunto de máquinas e notifique um administrador por e-mail caso uma máquina não responda a pings durante um período preestabelecido. Não defina internamente a lista de máquinas, o endereço de e-mail para notificação ou a quantidade de tempo usada para determinar o motivo para esta máquina não estar respondendo.

★ **E20.3** Experimente alterar a máscara de rede numa máquina em sua rede local.
Ela ainda funciona? Você consegue atingir tudo em seu site? Outras máquinas conseguem se comunicar com você? As difusões amplas funcionam (por exemplo, solicitações ARP ou pacotes de descoberta DHCP)? Explique suas descobertas. (Requer acesso root.)

★ **E20.4** Use o comando **traceroute** para descobrir caminhos de roteamento em sua rede.
a) Quantos saltos ele leva para deixar sua instalação?
b) Há algum roteador entre as máquinas para as quais você tem contas?
c) Você é capaz de encontrar algum gargalo?
d) O seu site tem várias rotas de acesso?

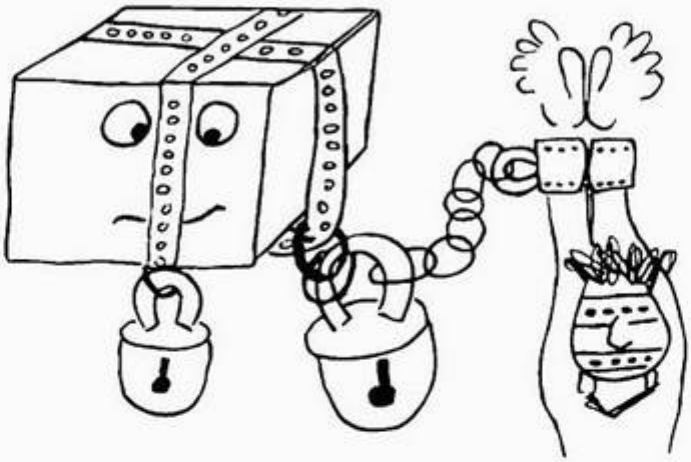
★★ **E20.5** Projete um MIB que inclua todas as variáveis que talvez você queira consultar ou configurar como administrador de sistemas Linux. Deixe maneiras para o MIB poder ser estendido para incluir esta nova variável de administração do sistema importante que você se esqueceu.

★★ **E20.6** Use o comando **tcpdump** para capturar tráfego que ilustre os seguintes protocolos. Para sessões TCP, inclua e indique os pacotes inicial e final. Apresente a saída do **tcpdump** organizada e bem formatada. (Requer acesso root.)
a) ARP
b) Solicitação de eco e resposta ICMP
c) SMTP
d) FTP e FTP-DATA
e) DNS (domínio chamado)
f) NFS

★★ **E20.7** Configure gráficos MRTG que ilustrem os pacotes transmitidos de e para um roteador local. Este projeto exige que um pacote SNMP consulte o roteador; você tem de conhecer a string comunitária somente para leitura do roteador.

www.snow666.blogspot.com
Snow666

21 Segurança



Em 1988, o mundo da tecnologia dos computadores e redes entrou numa nova era quando o "Internet Worm" de Robert Morris, Jr. desabou sobre a humanidade. Antes desse evento, a Internet (e o mundo da computação em geral) vivia na "idade da inocência". Segurança era um tópico que os administradores pensavam em grande parte no sentido "o que aconteceria se". Um grande incidente de segurança normalmente consistia de algo como um usuário ganhar acesso administrativo para ler a correspondência de um outro usuário, normalmente somente para provar que ele era capaz.

Na realidade, o worm de Morris causou poucos danos reais e aumentou em muito a consciência sobre segurança na Internet. Mais uma vez, fomos penosamente relembrados de que boas cercas fazem bons vizinhos. Surgiu uma série de ferramentas excelentes para serem usadas por administradores de sistemas (bem como uma organização formal para tratar de incidentes desta natureza) como conseqüência deste fato.

Infelizmente, como evidenciado pela pesquisa "CSI/FBI Computer Crime Survey", 91% das organizações relataram brechas de segurança nos 12 meses anteriores. É interessante notar que essas estatísticas alarmantes não se devem à ausência completa de tecnologia de segurança instalada, já que 95% dos que responderam à pesquisa relataram que usavam ferramentas deste tipo como os firewalls comercializados existentes. Esses dados ao menos nos ensinam que os firewalls não são por si só uma proteção adequada contra intrusos.

Em geral, a segurança não é algo que você pode adquirir num produto ou num serviço oferecido por terceiros. Os serviços e produtos comerciais podem fazer parte de uma solução para o seu site, mas eles não são uma panacéia. Alcançar um nível de segurança aceitável em seu site requer uma enorme dose de paciência, vigilância e persistência – não somente por parte de você e outros administradores como também de toda a sua comunidade de usuários e de gerentes. Como administrador de sistemas, você tem de assumir pessoalmente a responsabilidade de garantir que seus sistemas são seguros e que seus usuários estão adequadamente treinados. Você deve estar familiarizado com as tecnologias de segurança atuais, monitorar ativamente listas de discussão sobre segurança e contratar pessoal especializado em segurança de sistemas para ajudá-lo quando os problemas excederem os seus conhecimentos.

21.1 O Linux É Seguro?

Não. O Linux não é seguro. Assim como nenhum outro sistema operacional que se comunica via rede. Para ter segurança total, absoluta e sem brechas, então você precisará de um espaço enorme[1] entre o seu computador e qualquer outro dispositivo (e justificadamente você provavelmente também precisará trancar o seu computador numa sala especial que bloqueie radiações eletromagnéticas conforme projetada pelo pessoal aposentado da inteligência soviética). Que graça há nisso?

---

1. Obviamente, o advento da tecnologia de redes wireless introduz todo um novo conjunto de problemas.

www.snow666.blogspot.com
Snow666

Há algumas medidas que você pode tomar para fazer com que seu sistema se torne um pouco mais resistente a ataques. Mesmo assim, várias falhas básicas no modelo Linux garantem que você jamais será capaz de atingir o nirvana da segurança:

- Como o UNIX, o Linux é otimizado para a conveniência e não faz a segurança fácil ou natural. A filosofia do Linux enfatiza a fácil manipulação de dados em um ambiente de rede e multiusuário.
- A segurança do Linux é efetivamente binária: ou você é um usuário impotente ou então é root. Recursos Linux como a execução setuid tendem a conferir total poder de uma só vez. Os menores lapsos na segurança podem comprometer sistemas inteiros.
- O Linux é desenvolvido por uma grande comunidade de programadores. Eles variam em nível de experiência, atenção a detalhes e conhecimento sobre o sistema Linux e suas interdependências. Como conseqüência, até mesmo os mais bem intencionados e incríveis novos recursos podem introduzir enormes brechas na segurança. Por outro lado, como o código-fonte do Linux se encontra disponível a qualquer um, milhares de pessoas podem examinar minuciosamente cada linha de código em busca de uma possível ameaça em termos de segurança. Acredita-se amplamente que esse tipo de arranjo resulte significativamente em melhor segurança do que sistemas operacionais fechados, em que um número limitado de pessoas tem a oportunidade de examinar cuidadosamente o código em busca de brechas de segurança.

Diversas brechas de segurança conhecidas no Linux jamais serão corrigidas e outras foram corrigidas em algumas versões, mas não em todas. Além disso, muitas instalações se encontram uma versão ou duas atrás, seja porque a localização seja muito problemática seja pelo fato deles não possuírem um contrato de manutenção de software com o seu fornecedor. Quando uma correção para uma falha de segurança é disponibilizada, a oportunidade para os hackers atacarem não desaparece da noite para o dia.

Pode parecer que a segurança no Linux devesse aumentar gradualmente ao longo do tempo, já que os problemas relacionados com segurança são descobertos e corrigidos, mas, infelizmente, parece que este não é o caso. O software de sistema está crescendo e se tornando cada vez mais complicado, os hackers estão ficando melhores e mais bem organizados e os computadores estão se conectando cada vez mais e mais intimamente na Internet. A segurança é uma batalha contínua que jamais pode ser realmente vencida. Lembre-se também de que

$$\text{Segurança} = \frac{1}{(1{,}072)(\text{Conveniência})}$$

Quanto mais seguro for o seu sistema, mais miseráveis você e seus usuários tendem a ser.

21.2 Segurança no Linux, a Versão CliffsNotes

Este capítulo discute uma ampla gama de questões relativas à segurança. Idealmente, você deveria executar todas elas dentro de seu ambiente. Entretanto, caso não tenha muito tempo ou paciência eis as seis questões mais importantes a serem consideradas, mais algumas regras extras de acordo com as quais se guiar. (A maioria dos administradores deve realmente digerir o conteúdo deste capítulo todo, provavelmente mais de uma vez.)

Filtragem de pacotes

Se estiver conectando um sistema Linux a uma rede com acesso à Internet, você deve *obrigatoriamente* ter um roteador para filtragem de pacotes ou firewall entre o sistema Linux e o mundo externo. Como alternativa, você pode configurar a filtragem de pacotes usando **iptables** no próprio sistema Linux (discutido na seção 21.11). Seja qual for a implementação adotada, o filtro de pacotes deve deixar passar somente tráfego destinado a serviços essenciais que você deseja especificamente disponibilizar por meio do sistema Linux.

Serviços desnecessários

Várias versões do Linux diferem muito em quais serviços de rede estão ativos como parte da instalação-padrão. Fica a seu critério examinar os serviços que estão habilitados em seu sistema e desativar qualquer um deles que não seja absolutamente necessário. Examinar o conteúdo de **/etc/inetd.conf** (ou, no Red Hat, os arquivos contidos em **/etc/xinetd.d**) deve ser seu primeiro passo.

Correções de software

Todos os principais distribuidores Linux lançam regularmente pacotes de software relacionados com segurança, normalmente algumas vezes por mês. Você *tem de* estar vigilante em relação a correções para sua versão de Linux (e

Snow666

para quaisquer pacotes de software que esteja usando) e instalá-los imediatamente. Tenha em mente que assim que uma correção estiver disponibilizada, os "caras maus" já saberão da brecha há semanas e talvez já seja tarde demais.

Backups

Coloque aquele array RAID e bem longe do CPD, meu caro. Você *deve obrigatoriamente* realizar backups regularmente de todos os seus sistemas para poder efetivamente recuperá-los depois de um incidente de segurança, caso algum ocorra. Nenhum nível de tecnologia de espelhamento, RAID ou "hot standby" elimina a necessidade dos backups. Informações sobre como realizar backups são fornecidas no Capítulo 10.

Senhas

Somos pessoas simples com regras simples. Eis uma delas: toda conta *deve obrigatoriamente* ter uma senha e deve ser algo que não possa ser adivinhado facilmente. Infelizmente, não é mais seguro enviar senhas reutilizáveis em forma de texto pela Internet, de modo que se você permite logins remotos em seu sistema, deve usar SSH ou algum outro sistema de autenticação (discutido a partir da seção 21.9).

Vigilância

Para garantir a segurança de seu sistema, você tem de monitorar regularmente (em geral, todos os dias) sua saúde, conexões de rede, tabela de processos e estado geral. Os problemas de segurança costumam começar pequenos e crescem rapidamente, de maneira que quanto antes você identificar uma anomalia, melhor você se sairá.

Filosofia geral

A segurança efetiva de um sistema tem suas origens no bom senso e é muito parecido com o ato de lidar com uma infestação de ratos em sua casa. Eis algumas regras básicas que você deve usar:

- Não deixe coisas que provavelmente sejam interessantes para os ratos em cima da mesa durante a noite. Queijo e pasta de amendoim são excelentes chamarizes de ratos.
- Não disponibilize lugares dentro da casa para os ratos construírem seus ninhos. Pilhas de roupas sujas espalhadas pelo chão são bons ninhos.
- Coloque ratoeiras próximo às paredes onde você normalmente vê ratos.
- Verifique as ratoeiras diariamente para reabastecê-las com iscas e jogar fora ratos esmagados. Ratoeiras cheias não capturam ratos e elas fedem.
- Evite usar venenos comerciais para atrair e matar ratos para lidar com a situação. Estes podem deixá-lo com ratos mortos próximo à sua parede ou matar o seu cachorro. As velhas e tradicionais ratoeiras são mais eficientes.
- Arranje um gato!

Você pode adotar essas mesmas regras (bem, ligeiramente modificadas) para assegurar seus sistemas Linux. Eis como você deve reescrevê-las:

- Não coloque arquivos em seus sistemas que provavelmente possam ser interessantes para hackers ou para funcionários abelhudos. Segredos de negócios, arquivos pessoais, folhas de pagamento, resultados de eleições etc. têm de ser tratados com cuidado se estiverem on-line. Assegurar tais informações com criptografia lhe dará um grau de segurança bem maior do que simplesmente tentar impedir que usuários desautorizados acessem os arquivos que contêm as guloseimas.
  A política de segurança de sua instalação deve especificar com que nível de sigilo as informações são tratadas. Veja o Capítulo 29, *Políticas e Diretrizes*, e a RFC2196 (*Site Security Handbook*) para algumas sugestões.
- Não forneça locais para os hackers construírem ninhos em seu sistema. Os hackers normalmente invadem um sistema e depois o utilizam como uma base de operações para entrar em outros. Diretórios FTP anônimos em que qualquer um tem direito de gravação, contas de grupos e contas com senhas escolhidas de qualquer jeito favorecem ninhos de hackers.
- Configure armadilhas para detectar invasões e tentativas de invasão. Ferramentas como **tripwire**, **tcpd** e **crack** (descritas na seção 21.8) podem ajudar a mantê-lo longe de possíveis problemas.
- Monitore os relatórios gerados por essas ferramentas de segurança. Um problema secundário que é ignorado em um relatório pode crescer até se transformar numa catástrofe na oportunidade em que o próximo relatório for gerado.

www.snow666.blogspot.com
Snow666

- Aumente seus próprios conhecimentos sobre segurança de sistemas Linux. Know-how tradicional, preparação dos usuários e bom-senso são as partes mais importantes do plano de segurança de uma instalação. Traga especialistas de fora para preencher lacunas, mas somente sob sua aprovação e acompanhamento de perto.
- Fique na espreita observando atividade não usual. Investigue qualquer coisa que lhe pareça estranho, como mensagens de log estranhas ou alterações na atividade de uma conta (mais atividade, atividade em horários estranhos ou quem sabe atividade enquanto o dono da conta está de férias).

21.3 Como a Segurança É Comprometida

As seções a seguir discutem alguns problemas comuns na segurança de sistemas Linux e as medidas contra seus efeitos. Mas antes de partirmos para os detalhes, devemos ter uma visão mais genérica de como problemas reais de segurança tendem a ocorrer. A maioria dos lapsos de segurança se origina de um dos seguintes tipos de problemas:

- **Prudência não confiável:** Os usuários (e administradores também) de um sistema computacional normalmente são os elos mais fracos na cadeia de segurança. Por exemplo, a America Online costumava ser notória por viver infestada por hackers que se faziam passar por funcionários da AOL. Os hackers enviam um e-mail para vítimas em potencial solicitando a elas para enviar de volta suas senhas como parte de "teste de sistema" ou "para verificar a sua conta". Usuários não sofisticados normalmente não sabiam (e ainda não sabem) nada melhor do que reclamar.
  Há variações incontáveis deste estratagema. Como administrador, parte de seu papel inclui ensinar aos usuários sobre hábitos de segurança apropriados. Muitos usuários ainda são novatos na Internet e normalmente não fazem idéia do número de esquemas fraudulentos e aberrações que estão soltos por aí. Diga a eles como selecionar e defender senhas boas, como proteger seus trabalhos e como não conversar com estranhos. Certifique-se de que suas instruções abranjam também comunicação que não seja via e-mail; um telefone também pode ser o melhor amigo de um hacker.
- **Bugs de software:** Ao longo dos anos, um número incontável de bugs referentes à segurança foram descobertos em softwares do Linux (inclusive software de terceiros, sejam eles produtos comerciais ou gratuitos). Explorando erros de programação sutis ou dependências de contexto, os hackers foram capazes de manipular o Linux para fazer o que bem entendessem. O que você pode fazer como administrador para impedir isso? Muito pouco, pelo menos enquanto um bug não tiver sido identificado e tratado numa correção. Manter-se atualizado em relação às correções e aos boletins de segurança é uma parte importante das tarefas da maioria dos administradores.
- **Portas abertas:** Muitos pacotes e software podem ser configurados de maneira segura ou não tão segura. Infelizmente, normalmente a forma não tão segura é a padrão. Os hackers freqüentemente ganham acesso explorando recursos de software que seriam considerados úteis e convenientes em circunstâncias menos insidiosas: contas sem senhas, discos compartilhados com todo o mundo e logins remotos equivalentes entre as máquinas, só para citar alguns. Uma das partes mais importantes para assegurar um sistema é simplesmente ter certeza de que você inadvertidamente não deixou um tapete de boas-vindas para hackers.

Problemas que caem na última dessas categorias são os mais fáceis de serem localizados e corrigidos, embora haja potencialmente um bocado deles e nem sempre é óbvio o que verificar.

*Consulte a seção 6.1 para obter maiores informações sobre o arquivo **passwd**.*

21.4 Problemas de Segurança nos Arquivos /etc/passwd e /etc/shadow

Um controle ineficiente das senhas é um ponto fraco comum em termos de segurança do sistema. O conteúdo dos arquivos **/etc/passwd** e **/etc/shadow** determinam quem pode entrar e o que pode fazer assim que tiverem entrado no sistema. Esses arquivos são a primeira linha de defesa de um sistema contra intrusos. Eles devem ser escrupulosamente mantidos e isentos de erros, riscos de segurança e bagagem histórica.

Verificação e seleção de senhas

É importante verificar continuamente (de preferência diariamente) se todo login tem uma senha. Entradas no arquivo **/etc/shadow** (ou no arquivo **/etc/passwd** se sua instalação não usa senhas-sombra) que descrevem pseudo-usuários como "daemon" que possuem arquivos, mas jamais entram no sistema, devem ter um asterisco (*) em seus campos de senhas criptografadas. O asterisco não coincidirá com nenhuma senha e, portanto, impedirá o emprego da conta.

www.snow666.blogspot.com
Snow666

Existem diversos pacotes de software especializados para checar **/etc/shadow** em busca de problemas de segurança, porém o comando[2]

```
perl -F: -ane 'print if not $F[1];' /etc/shadow
```

já é o suficiente para localizar senhas nulas. Um script que realiza essa verificação e envia e-mails para você com os resultados pode ser executado por meio de **cron**. Você pode acrescentar segurança extra criando um script que compare (aplique **diff**s ao) o arquivo **passwd** em relação a uma versão do dia anterior e lhe envie e-mails apontando quaisquer diferenças. A seguir você poderá verificar se quaisquer modificações são legítimas.

/etc/passwd e **/etc/group** têm de oferecer acesso de leitura a todo mundo, mas de gravação somente para root. Se seu sistema possui um arquivo **/etc/shadow**, o "mundo" não deve ter nenhum acesso a ele.

O Linux permite que os usuários escolham suas próprias senhas e embora isto seja uma grande conveniência, também leva a diversos problemas de segurança. Ao dar logins a seus usuários, você também deve dar instruções a eles sobre como escolher uma boa senha. Diga a eles para não usarem seus nomes ou iniciais, o nome do filho ou da esposa ou qualquer outra palavra que possa ser encontrada num dicionário. As senhas derivadas de dados pessoais como números de telefone ou endereços também podem ser descobertas facilmente.

As senhas devem ter pelo menos oito caracteres de comprimento e devem incluir números, pontuação ou mudanças na caixa das letras. Palavras sem sentido, combinações de palavras simples ou as primeiras letras de uma frase memorável formam as melhores senhas. Obviamente, "memorável" é bom, porém "tradicional" é perigoso. Construa a sua própria frase. Os comentários da seção *Selecionando uma senha root*, do Capítulo 3, também se aplicam a senhas de usuários.

Em alguns poucos sistemas (dependendo das bibliotecas de autenticação que você esteja usando), somente os oito primeiros caracteres de uma senha são significativos. Podem ser introduzidos mais caracteres, porém aqueles acima do oitavo serão silenciosamente ignorados. Veja a seção 6.1 para detalhes.

PAM: spray culinário ou prodígio de autenticação?

O PAM (Pluggable Authentication Modules, módulos de autenticação plugáveis) foi originalmente inventado pela Sun como uma forma flexível de autenticar usuários. Por vários anos, a autenticação no ambiente UNIX era mera questão de associar usuários a suas entradas no arquivo **/etc/passwd**. Mais tarde, a necessidade de uma maior segurança e de suporte para uma ampla gama de mecanismos de autenticação (como cartões inteligentes — "smart cards", por exemplo) criou uma necessidade de uma abordagem mais flexível.

O Linux-PAM vem com o Red Hat, SuSE e Debian e não está relacionado com a implementação atual da Sun do padrão PAM. O conceito é simples: programas que requerem autenticação somente precisam saber que há um módulo disponível que irá fazer a autenticação para eles. O PAM é configurado de tal forma que esses módulos possam ser acrescentados, eliminados e reconfigurados a qualquer momento – não é necessário que os módulos sejam linkados (ou mesmo que existam) no momento em que um utilitário é compilado. Como resultado dessa arquitetura, o PAM se tornou uma ferramenta incrivelmente poderosa para administradores de sistemas.

Os módulos PAM são configurados através de arquivos no diretório **/etc/pam.d**. Arquivos por serviço nesse diretório contêm entradas no formato

```
tipo-do-módulo opções-de-controle caminho-do-módulo argumentos
```

O campo *tipo-do-módulo* pode assumir os valores `auth`, `account`, `session` ou `password`. Uma entrada `auth` estabelece quem é o usuário e possivelmente concede associação de grupo. A tag `account` toma decisões não baseadas em autenticação, como acesso baseado em horários. Tarefas que precisem ser realizadas antes ou depois de um usuário ter recebido um serviço são implementadas com a tag `session`. Finalmente, a tag `password` é usada quando informações de autenticação (como, por exemplo, uma senha) são solicitadas ao usuário.

O campo *opções-de-controle* possui quatro valores possíveis: `required`, `requisite`, `sufficient` e `optional`. `required` e `optional` são mais comumente usados indicando, respectivamente, que um módulo tem de ser bem-sucedido para que a execução prossiga ou então que não importa se o módulo foi ou não bem-sucedido.

Consulte a seção 21.8 para obter maiores informações sobre ***crack****.*

Os terceiro e quarto campos são o nome de caminho e os argumentos para o objeto de módulo carregável dinamicamente. Se o primeiro caractere do caminho for /, parte-se da suposição que ele seja um caminho absoluto. Caso contrário, o conteúdo do campo é anexado ao caminho-padrão, **/lib/security**.

2. Esse comando requer Perl 5 ou posterior.

Snow666

Por exemplo, adições a **/etc/pam.d/passwd** para habilitar o programa **passwd** para realizar verificação rígida de senhas usando um módulo PAM derivado da biblioteca **crack** ficaria parecido com o seguinte:

```
password required pam_cracklib.so retry=3
password required pam_pwdb.so use_authtok
```

Com essas linhas no lugar apropriado, o PAM verificará as novas senhas propostas pelos usuários em relação a um conjunto de regras e a um dicionário **crack**. (Esse ajuste de configuração requer a presença da biblioteca de sistema **libcrack** e também de um dicionário de sistema, **/usr/lib/cracklib_dict**.) O argumento `retry=3` especifica que o usuário terá três chances de introduzir uma senha forte. A linha contendo `use_authtok` conecta entre si as camadas do módulo de senha.

Dezenas de módulos PAM estão disponíveis. Você pode fazer um download de módulos especializados e respectiva documentação em www.kernel.org/pub/linux/libs/pam.

Senhas-sombra (shadow)

Tradicionalmente, cada linha em **/etc/passwd** é formada por sete campos. O segundo campo contém uma string que representa a senha criptografada do usuário. Já que **/etc/passwd** tem de poder ser lido por todo o mundo para comandos como **ls** funcionarem, a string de senha criptografada fica disponível para todos os usuários do sistema. Malfeitores podem criptografar palavras ou dicionários selecionados e comparar os resultados com as strings contidas em **/etc/passwd**. Se as strings criptografadas forem coincidentes, foi localizada uma senha.

Qual o nível de ameaça disso? Nos anos 80, havia pelo menos uma maneira rápida de decifrar senhas[3], mas hackers medianos tinham que se contentar em usar a rotina de biblioteca **crypt**[4] para criptografar palavras de dicionário para comparação. Uma máquina "rápida" dos anos 80 era capaz de realizar algumas centenas de criptografias por segundo. Em 1998, John Gilmore da Electronic Frontier Foundation e o especialista em criptografia Paul Kocher quebraram uma senha DES de 56 bits em 56 horas, usando uma pesquisa de força bruta. Propostas recentes sugerem que um computador de propósito especial de US$ 1 milhão seria capaz de descobrir qualquer senha DES de 56 bits em apenas algumas horas.

Esses resultados são atemorizantes e sugerem que acesso de usuários a strings de senhas criptografadas deve realmente ser restringido. A maneira-padrão para impor restrições é colocar senhas num arquivo separado que possa ser lido somente por root, deixando intacto o restante do arquivo **/etc/passwd**. O arquivo que contém as informações de senha em si é então chamado pelo arquivo de senhas-sombra, **/etc/shadow**. A maioria das distribuições Linux usa senhas-sombra em sua configuração-padrão e aquelas que não o fazem deveriam fazê-lo.

Consulte a seção 29.2 para alguns comentários adicionais sobre logins compartilhados.

Logins de grupos e logins compartilhados

Qualquer login que seja usado por mais de uma pessoa é sinal de más notícias. Os logins de grupos (como "guest" ou "demo") são terreno fértil para hackers se hospedarem. Não permita o seu uso em sua instalação.

Da mesma forma, não permita que usuários compartilhem logins com familiares ou amigos. Se o Joãozinho precisar de um login para poder operar em seu trabalho de ciências, dê um login a ele com este declarado propósito. É muito mais fácil tirar o login do Joãozinho quando ele abusar dele do que se livrar do papai e de sua conta, especialmente em instalações governamentais.

Na maioria delas, "root" é um login de grupo. Perigo! Recomendamos usar o programa **sudo** para controlar o acesso a poderes de root. Veja o Capítulo 3.

Expiração da senha

O sistema de senhas-sombra do Linux também permite que você force seus usuários a trocar de senhas periodicamente, através de um recurso conhecido como expiração de senha. Isso pode, à primeira vista, parecer uma boa idéia, mas possui vários problemas. Os usuários muitas vezes se tornam ressentidos em ter de trocar de senha e já que eles não querem se esquecer da nova senha, optam por algo simples que seja fácil de digitar e de lembrar. Muitos usuários alternam entre duas senhas cada vez que são forçados a modificá-las, debilitando o propósito do processo de expiração de senhas.

3. Evi Nemeth quebrou o intercâmbio de senhas Diffie-Hellman normalmente usado com DES em 1984, por meio de um supercomputador HEP. Embora se imagine que DES seja matematicamente seguro, os comprimentos curtos das senhas em uso comum oferecem relativamente pouca segurança.
4. Não confunda a rotina de biblioteca **crypt** com o comando **crypt** que usa um esquema de criptografia diferente e menos seguro.

Snow666

Consulte o Capítulo 3 para obter maiores informações sobre ***sudo****.*

Não obstante, as senhas devem ser alteradas regularmente, em especial a senha root. Uma senha root deve fluir facilmente de modo que possa ser digitada rapidamente e não seja descoberta por alguém que observe o movimento de seus dedos pelo teclado. Em nossa instalação a maioria das pessoas usa **sudo** em vez da senha root real, porém selecionamos cuidadosamente a senha root da mesma maneira.

Shells de usuários

Não use um script como o shell para login irrestrito (sem senha). Logins sem senha somente deveriam ser utilizados como um recurso para a execução de pequenos utilitários não interativos como **date**, **sync**, ou **lpq**.

Entradas root

O único aspecto que distingue o login root é o seu UID zero. Já que pode haver mais de uma entrada no arquivo **/etc/passwd** que use esse UID, pode haver mais de uma maneira de fazer o login como root.

Uma maneira comum para os hackers instalarem uma porta dos fundos assim que tenham obtido um shell root é editar novos logins root em **/etc/passwd**. Programas como **who** e **w** referem-se ao nome armazenado em **/var/run/utmp** em vez do UID que é proprietário do shell de login, de modo que eles não são capazes de expor os hackers que aparentam ser inocentes usuários mas que estão logados realmente como UID 0.

A defesa contra esse subterfúgio é um miniscript similar àquele usado para localizar logins sem senhas[5]:

```
perl -F: -ane 'print if not $F[2];' /etc/passwd
```

Esse script imprime quaisquer linhas contidas no arquivo **passwd** que possuam UIDs nulos ou 0. Você poderia adaptá-lo facilmente para localizar entradas com grupos suspeitos ou UIDs que são os mesmos daqueles de pessoas-chave dentro de sua empresa.

Você também deveria verificar a existência de entradas **passwd** que não possuam nome de usuário ou que tenham pontuação como nome de usuário. Essas entradas podem parecer sem sentido, porém elas normalmente permitirão que um hacker entre no sistema.

21.5 Programas Setuid

Programas que executam setuid, especialmente aqueles que executam setuid em root, são propensos a problemas de segurança. Os comandos setuid distribuídos com o Linux são teoricamente seguros; entretanto, brechas de segurança foram descobertas no passado e indubitavelmente serão descobertas no futuro.

A maneira mais garantida de minimizar o número de *problemas* relacionados com setuid é minimizar o número de *programas* setuid. Pense duas vezes antes de instalar um pacote de software que precise executar setuid e evite usar o recurso setuid em seu software caseiro. Scripts de shell setuid são especialmente aptos a provocar problemas de segurança. Pelo menos sob um shell comum, eles são automática e inteiramente inseguros. Os shells tendem a ser altamente personalizáveis, o que os torna alvos relativamente fáceis de serem enganados. Embora um shell gerado para executar um script não leia necessariamente os arquivos de configuração de shell, ele pode ser influenciado pelo ambiente do usuário, pelo conteúdo do diretório atual ou pela maneia pela qual o script é chamado.

Não há nenhuma regra que diga que os programas setuid tenham de ser executados como root. Se tudo o que você precisa fazer é restringir o acesso a um determinado arquivo ou banco de dados, você poderá adicionar um pseudo-usuário ao arquivo **passwd** cuja única razão para existência é ter como propriedade os recursos restritos. Siga as convenções de pseudo-usuários normais: use um UID baixo, coloque um asterisco no campo de senha e faça com que o diretório de origem do pseudo-usuário seja **/dev/null**.

Você pode desabilitar a execução de setuid e setgid em sistemas de arquivos individuais pelo uso da opção **-o nosuid** em **mount**. É uma boa idéia usar essa opção em sistemas de arquivos que contenham diretórios de usuários ou que sejam montados por meio de domínios administrativos menos confiáveis.

É útil fazer uma varredura periódica de seus discos em busca de novos programas setuid. Um hacker que rompeu a segurança de seu sistema irá, algumas vezes, criar um utilitário ou shell setuid privado para facilitar visitas repetidas. Algumas dessas ferramentas discutidas na seção 10.9 localizarão tais arquivos, mas você pode fazer o mesmo com **find**. Por exemplo,

```
/usr/bin/find / -user root -perm -4000 -print |
   /bin/mail -s "Setuid root files" netadmin
```

enviará uma lista de todos os arquivos setuid root para o usuário "netadmin".

5. Esse comando requer Perl 5 ou posterior.

www.snow666.blogspot.com
Snow666

21.6 Permissões de Arquivos Importantes

Muitos arquivos num sistema Linux devem ter permissões particulares se quisermos evitar problemas com segurança. Alguns distribuidores enviam seu software com permissões configuradas para seus próprios ambientes de desenvolvimento "amigáveis". Essas permissões talvez não sejam apropriadas para o seu caso.

O arquivo de dispositivo **/dev/kmem** permite acesso ao espaço de endereços do próprio kernel. Ele é usado por programas como **ps** que precisam examinar estruturas de dados do kernel. Esse arquivo deveria ter permissão de leitura somente para o proprietário e grupo, jamais pra todo o mundo. Programas que precisam acessar esse arquivo devem ser setgid para o grupo que possui o arquivo, normalmente kmem.

No passado, alguns sistemas foram distribuídos de forma descuidada com **/dev/kmem** podendo ser lido por todo o mundo. Trata-se de um grave problema de segurança, pois um programador competente pode procurar coisas como senhas não criptografadas nas estruturas de dados e buffers do kernel. Se seu sistema tem **/dev/kmem** aberto ao público, altere isto imediatamente. Se a alteração fizer com que qualquer programa pare de funcionar, torne esses programas setgid para o grupo que é proprietário de **/dev/kmem./etc/passwd** e **/etc/group** não devem estar disponíveis para gravação por todo o mundo. Eles devem ter root para o proprietário e modo 644. **/etc/shadow** deve ter modo 640 — nenhuma permissão para o "mundo". O grupo de todos esses arquivos deve ser configurado para algum grupo de sistema, normalmente root. O comando **passwd** executa setuid para root de modo que os usuários possam alterar suas senhas sem ter permissão de gravação sobre **/etc/passwd** ou **/etc/shadow**.

Consulte a seção 22.6 para obter maiores informações sobre como configurar um servidor FTP seguro.

Diretórios que são acessíveis através de FTP anônimo não devem ser graváveis por todos. Tais diretórios criam um ninho para os hackers distribuírem software copiado ilegalmente e outros arquivos sigilosos. Se você gerencia um repositório FTP que permite tais submissões, certifique-se de fazer uma triagem do diretório de submissões regularmente.

Configurar FTP anônimo normalmente envolve copiar um arquivo-esqueleto de senhas em **~ftp/etc/passwd** de modo que **ls** funcionará corretamente. Certifique-se de eliminar as strings de senha criptografadas.

Arquivos de dispositivos para partições de disco rígido é uma outra fonte potencial para problemas. Ter permissão de leitura ou de gravação num arquivo de dispositivo de disco é basicamente a mesma coisa que ter permissão de leitura ou gravação em todo arquivo no sistema de arquivos que ele representa. Somente root deve ter tanto permissão de leitura quanto permissão de gravação. O proprietário do grupo algumas vezes recebe permissão de leitura para facilitar os backups, mas não deve haver permissões para o "mundo".

21.7 Problemas de Segurança Variados

As seções a seguir apresentam alguns tópicos variados relacionados à segurança. A maioria cai na categoria de recursos que são úteis a você como administrador de sistemas ou recursos impróprios que podem fornecer material para instalação de hackers se não forem verificados constantemente.

Veja o Capítulo 11 para obter maiores informações sobre syslog.

Registro de eventos remoto

O recurso syslog permite que informações de log de kernel, bem como de processos de usuário, possam ser encaminhadas a um arquivo, uma lista de usuários ou um outro host de sua rede. Considere a possibilidade de configurar um host seguro que atue como uma máquina de log central e imprima violações de segurança (o recurso auth) numa impressora de linha antiga. Essa precaução impedirá que hackers encubram seus rastros regravando ou apagando arquivos de log.

Terminais seguros

O Linux pode ser configurado para restringir logins root restritos a terminais "seguros" específicos. É uma boa idéia desabilitar logins root através de canais como modems de conexão discada. Freqüentemente, pseudoterminais de rede são configurados para desabilitar login root.

Os canais seguros são especificados como uma lista de dispositivos TTY no arquivo de configuração **/etc/securetty**. Também é possível restringir logins não-root para determinadas localizações com entradas no arquivo **/etc/security/access.conf** ou para horários particulares com entradas no arquivo **/etc/security/time.conf**.

Snow666

/etc/hosts.equiv and ~/.rhosts

Os arquivos **hosts.equiv** e **~/.rhosts** definem hosts como administrativamente "equivalentes" uns aos outros, permitindo aos usuários fazerem o login (através de **rlogin**) e copiar arquivos (com **rcp**) entre máquinas sem digitar suas senhas.[6] O emprego desse recurso já foi comum durante os tempos de festa do UNIX, mas todo o mundo acordava eventualmente com uma horrível dor de cabeça e se dava conta de que não era uma boa idéia.

Recomendamos que **rshd** e **rlogind**, os processos de servidor que lêem **.rhosts** e **hosts.equiv**, sejam desabilitados. Nos dias de hoje, **telnetd** também deve ser desabilitado. Na maioria dos sistemas, desabilitamos esses servidores colocando símbolos de comentário em **/etc/inetd.conf** (ou, no Red Hat, desabilitando-os nos arquivos **/etc/xinetd.d**). Assim que os daemons de servidor tiverem sido desabilitados, o host não poderá mais ser acessado via **telnet**, **rlogin**, **rsh** ou **rcp**. Entretanto, a funcionalidade desses comandos pode ser substituída com equivalentes mais seguros como SSH; veja a seção 21.9.

Algumas das substituições para **rlogin** (inclusive SSH!) prestam atenção a **.rhosts** e a **/etc/hosts.equiv** caso eles não sejam configurados apropriadamente. Para maior segurança, você poderá criar o arquivo **/etc/hosts.equiv** e um arquivo **~/.rhosts** para cada usuário (inclusive root) como um arquivo de comprimento zero e que não possa ser gravado. É mais fácil verificar qual era o estado de um arquivo às 3 da manhã se ele existir e não for tocado do que verificar o estado de um arquivo inexistente. Essa distinção pode ser crucial quando você estiver rastreando intrusos e suas tentativas de comprometer seu sistema.

rexecd e tftpd

rexecd é mais um daemon de execução de comandos remotos. Ele é o servidor para a rotina de bibliotecas **rexec**. Solicitações enviadas para **rexecd** incluem uma senha na forma de texto puro, de maneira que qualquer um que esteja na escuta da rede pode descobrir senhas e ganhar acesso ao sistema-alvo. Desabilite esse daemon.

tftpd é um servidor para o Trivial File Transfer Protocol, um protocolo fácil de ser implementado que algumas vezes é usado para fazer o download de firmware ou código de boot em dispositivos de rede. Pelo fato de permitir que máquinas na rede solicitem arquivos de seu disco rígido, ele se torna uma brecha de segurança em potencial. É melhor deixá-lo desabilitado caso não esteja usando esse recurso.

fingerd

finger é um comando Linux que imprime um pequeno relatório sobre um usuário em particular:

```
% finger evi
Login: evi                  Name: Evi Nemeth
Directory: /home/evi        Shell: /usr/bin/tcsh
Last login Thu Nov 29 12:32 (MST) on pts/0 from natted.sendmail.com
No Mail.
No Plan.
```

Sem o uso de argumentos, **finger** imprime uma síntese de todos os usuários que estão logados no sistema.

Quando suportado pelo daemon **fingerd** num host remoto, **finger** também pode ser executado na forma **finger** *usuario@host* ou simplesmente **finger** *@host*. Infelizmente, as informações retornadas são potencialmente úteis para os hackers, de modo que recomendamos que você desabilite **fingerd** em **/etc/inetd.conf** (ou no arquivo **fingerd** em **/etc/xinetd.d** em Red Hat).[7]

Veja o Capítulo 18 para obter maiores informações sobre NIS.

Segurança e NIS

Além do título desta seção, essas palavras jamais devem ser usadas juntas. O NIS (Network Information Service, serviço de informações de rede; anteriormente denominado Yellow Pages, páginas amarelas) é uma ferramenta de distribuição de banco de dados da Sun que muitas instalações usam para manter e distribuir arquivos como **/etc/group**, **/etc/passwd** e **/etc/hosts**. Infelizmente, sua natureza intrínseca de "fácil acesso a informações" a transforma numa apetitosa isca para hackers. Uma versão posterior do NIS, chamada NIS+, faz uma tentativa ineficiente de corrigir problemas de segurança do NIS. É melhor você não executar nenhuma das formas do NIS em sua instalação.

---

6. Esses arquivos também são usados pelo software de impressão para autorizar acesso a impressoras. Veja o Capítulo 24, *Impressão*, para maiores detalhes.
7. Também é digno de nota que uma série de bugs relacionados com segurança foram descobertos em **fingerd** ao longo dos anos, o que não é comum para um programa tão simples como este.

www.snow666.blogspot.com
Snow666

Uma maneira mais segura e confiável de distribuir esses arquivos é criar um login como "netadmin" e colocar as cópias mais recentes destes arquivos em **~netadmin**. A seguir, você poderá executar um script por meio de **cron** em cada cliente para realizar um **scp**, uma verificação de sanidade e instalar os arquivos. Veja a seção 21.9 para obter maiores informações sobre SSH, do qual **scp** faz parte.

Segurança e NFS

Veja a seção 17.1 para obter maiores informações sobre segurança NFS. Você pode usar **showmount -e** para verificar que sistemas de arquivos estão sendo exportados e para quem. Cada sistema de arquivos exportado deve ter uma lista de acesso, e todos os nomes de hosts devem ser totalmente qualificados.

*Veja o Capítulo 19 para obter maiores informações sobre **sendmail**.*

Segurança e sendmail

sendmail é um sistema de rede enorme, cuja grande parte é executada como root. O resultado disso é que ele tem sido sujeito a ataques de hackers e numerosas vulnerabilidades têm sido expostas ao longo do tempo. Certifique-se de que esteja usando a versão mais atualizada de **sendmail** em todos os seus sistemas. Já que problemas de segurança é uma das questões mais prováveis a incitar novas versões de software, é provável que todas as versões de **sendmail** exceto a mais atual os tenha.

Detalhes específicos sobre segurança no **sendmail** são vistos no capítulo sobre o sendmail na seção 19.11, e informações sobre a segurança de sua versão mais atual se encontra disponível em www.sendmail.org.

Veja o Capítulo 10 para obter maiores informações sobre backups.

Segurança e backups

Backups regulares do sistema é uma parte essencial do plano de segurança de qualquer instalação. Certifique-se de que todas as partições sejam regularmente colocadas em fita e que alguns backups sejam armazenados fora da instalação. Se ocorrer um incidente significativo de segurança, você terá um ponto de verificação não contaminado para poder restaurar.

Backups também podem ser uma ameaça à segurança. Já que qualquer um pode ler o conteúdo de uma fita assim que ela estiver montada numa unidade de fita, você tem de manter todas as fitas de backup trancadas num lugar seguro.

Cavalos de Tróia

Cavalos de tróia são programas que não são o que parecem. Um exemplo de um cavalo de tróia foi um programa chamado **turkey** distribuído na Usenet há muito tempo. O programa dizia que desenharia a figura de um peru na sua tela, porém na verdade ele apagava arquivos de seu diretório de usuário.

Dado o número de aventuras relacionadas à segurança que a comunidade Linux vem assistindo ao longo dos últimos anos, é impressionante o pequeno número de incidentes relacionados a cavalos de tróia. Na realidade, não estamos cientes de uma única ocorrência documentada de um programa que

- tivesse o sentido de ter algum propósito útil,
- não fosse distribuído como parte de um sistema operacional,
- fosse fornecido na forma de código-fonte e que
- estivesse amplamente disponível

que contivesse código mal-intencionado ou que intencionalmente frustrasse a segurança de um sistema. Não nos interprete mal: estamos certos de que isso já aconteceu. Porém o risco para o administrador típico é muito baixo.

O crédito para esse estado de coisas se deve, em grande parte, à comunidade da Internet. Problemas de segurança óbvios tendem a descobertos rapidamente e discutidos amplamente. Pacotes mal-intencionados não ficam disponíveis por muito tempo em servidores Internet bem conhecidos.

Pode estar certo de que qualquer software que tenha sido descoberto como mal-intencionado será discutido amplamente na Internet. Caso queira fazer uma rápida verificação antes de instalar algo, digite o nome do pacote de software em seu mecanismo de busca favorito.

21.8 Ferramentas de Segurança Poderosas

Algumas das tarefas de impedir a formação de alojamentos para hackers mencionadas nas seções anteriores podem ser automatizadas com ferramentas gratuitas. Eis algumas delas que você deve analisar.

Snow666

nmap: varre portas de rede

nmap é um varredor de portas de rede. Sua principal função é verificar um conjunto de hosts de destinos para ver quais portas TCP e UDP possuem servidores escutando nelas.[8] Já que a maioria dos serviços de rede está associada a números de portas "bem conhecidos", essas informações dizem um bocado sobre o software que uma máquina está executando.

Executar **nmap** é uma excelente maneira de descobrir como um sistema se parece para alguém que está tentando invadir este sistema. Eis, por exemplo, um relatório gerado por uma máquina comum e relativamente insegura:

```
% nmap -sT host1.uexample.com
Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com, www.insecure.org/nmap/)
Interesting ports on host1.uexample.com (10.10.2.1):

Port  State Protocol Service
7     open  tcp      echo
9     open  tcp      discard
13    open  tcp      daytime
19    open  tcp      chargen
21    open  tcp      ftp
23    open  tcp      telnet
25    open  tcp      smtp
...
513   open  tcp      login

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

O argumento **-sT** solicita a **nmap** para tentar e conectar a cada porta TCP do host de destino segundo a maneira tradicional.[9] Assim que uma conexão tiver sido estabelecida, **nmap** se desconecta imediatamente, o que é descortês, porém não prejudicial a um servidor de rede configurado de forma apropriada.

Do exemplo anterior, podemos ver que host1.uexample.com está executando vários servidores que têm sido historicamente associados a problemas de segurança: **ftpd** (`ftp`), **rlogind** (`login`) e, provavelmente, **sendmail** (`smtp`). Várias linhas de ataque em potencial ficaram claras.

A coluna State na saída gerada por **nmap** mostra "open" para portas com servidores, "unfiltered" para portas sem servidores e "filtered" para portas que não podem ser sondadas devido à interposição de um firewall. Portas não filtradas ("unfiltered") são o caso típico e normalmente não são exibidas a menos que haja relativamente poucas delas. Eis, por exemplo, um dump de um servidor Web comercial mais seguro, www.aexample.com:

```
% nmap -sT www.aexample.com
Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com, www.insecure.org/nmap/)
(Not showing ports in state: filtered)

Port  State       Protocol Service
53    unfiltered  tcp      domain
80    open        tcp      http
179   unfiltered  tcp      bgp
443   open        tcp      https

Nmap run completed -- 1 IP address (1 host up) scanned in 122 seconds
```

Nesse caso, fica claro que o host está configurado para manipular somente tráfego Web. Um firewall bloqueia o acesso a outras portas. Tráfego DNS e BGP tem permissão, mas nenhum servidor está rodando para recebê-lo. Idealmente, o firewall nessa instalação deveria bloquear tráfego a todos os serviços não utilizados (como BGP e DNS nesse caso) de modo que essas portas não possam ser manipuladas por hackers para outros propósitos.

Além de sondagens diretas em TCP e UDP, **nmap** também possui um repertório de maneiras dissimuladas de sondar portas sem realmente iniciar uma conexão. Na maioria dos casos, essas sondas enviam pacotes que se parecem como tendo sido provenientes do meio de uma conversação TCP (e não do início) e aguardam que sejam envia-

---

8. Conforme descrito no Capítulo 13, uma porta é um canal de comunicações numerado. Um endereço IP identifica uma máquina inteira e um endereço IP mais um número de porta identifica um servidor específico ou uma conversação de rede nesta máquina.
9. Na verdade, somente as portas privilegiadas (aquelas com números abaixo de 1.024) e as portas bem conhecidas são verificadas na situação-padrão. Use a opção **-p** para especificar explicitamente o intervalo das portas a serem varridas.

www.snow666.blogspot.com
Snow666

dos de volta pacotes de diagnóstico. As sondas indetectáveis podem ser capazes de passar por um firewall ou de evitar a detecção por um monitor de segurança de rede na busca por varreduras de portas. Se a sua instalação usa um firewall (veja *Firewalls*, mais à frente), é uma boa idéia sondá-lo com esses modos de varredura alternativos para ver o que acontece.

nmap possui a capacidade mágica e útil de adivinhar que sistema operacional está sendo usado por um sistema remoto analisando os detalhes de sua implementação TCP/IP. A opção **-O** ativa esse comportamento. Por exemplo:

```
% nmap -O disaster mrhat lollipop
Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com, www.insecure.org/nmap/)

Interesting ports on disaster.xor.com (192.108.21.99):
...
Remote operating system guess: HP-UX 11.00

Interesting ports on mrhat.xor.com (192.108.21.2):
...
Remote operating system guess: BSDI 4.0

Interesting ports on lollipop.xor.com (192.108.21.48):
...
Remote operating system guess: Solaris 2.6 - 2.7

Nmap run completed -- 3 IP addresses (3 hosts up) scanned in 5 seconds
```

Esse recurso pode ser muito útil para extrair um inventário de uma rede local. Infelizmente, ele também é útil para os hackers, que podem basear seus ataques em fraquezas conhecidas do sistema operacional que está sendo usado pela máquina que eles desejam atacar. Tenha em mente também que a maioria dos administradores normalmente não aprecia o seu empenho em ficar varrendo suas redes e apontando suas vulnerabilidades, a despeito de sua boa intenção. *Jamais* execute **nmap** na rede de terceiros sem a permissão de um de seus administradores de rede.

ndiff: crie linhas de referência baseadas em nmap e procure encontrar alterações suspeitas

Uma janela para o passado pode ser o melhor amigo de um administrador de sistemas quando se trata de identificar brechas de segurança. Baseando-se na funcionalidade de **nmap**, **ndiff** fornece uma maneira de procurar mudanças interessantes nos estados das portas e hosts visíveis. O comando **ndiff** em si é muito simples: ele compara duas varreduras **nmap** e exibe as diferenças. Quase da mesma maneira que o sistema **tripwire** discutido posteriormente ainda neste capítulo registra uma imagem-base de seu sistema de arquivos e depois o alerta quando houver mudanças em permissões de arquivos ou checksums, o sistema **ndiff** registra uma linha de referência das portas e hosts que respondem dentro de sua rede.

Dois comandos distribuídos com **ndiff** ajudam a facilitar todo esse processo: **ngen** e **nrun**. **ngen** permite que você crie uma linha de referência baseada em expectativas (como você gostaria que sua rede ficasse). **nrun** executa **nmap**, armazena os resultados em formato de máquina e depois aplica **ndiff**s e os compara com uma linha de referência real ou imaginária. **nrun** pode ser executado facilmente por meio de **cron** de forma regular.

A distribuição **ndiff** inclui um conversor HTML para os resultados, de modo que você possa colocá-los facilmente em uma página Web (não numa que possa ser acessada por todo o mundo, esperamos, a menos que o seu intento seja o de publicar um guia para os hackers para sua instalação). Você pode obter maiores informações ou fazer o download de **ndiff** em www.vinecorp.com/ndiff.

SAINT: verifica a eventual existência de vulnerabilidades de sistemas ligados em rede

SAINT é uma versão atualizada do SATAN, um verificador da segurança de redes lançado em 1995 em meio a preocupações excessivas sobre como ele provocaria o fim do mundo. O SATAN original foi criado por Dan Farmer e Wietse Venema; o SAINT agora é mantido pela World Wide Digital Security, Inc., de cujo site Web pode ser feito o seu download (www.wwdsi.com).[10]

Assim como o **nmap**, o SAINT sonda computadores numa rede procurando descobrir que servidores eles estão usando naquele momento. Mas diferentemente de **nmap**, SAINT sabe bastante em relação aos programas de

10. Atualmente (novembro de 2003) SAINT está disponível em uma versão comercial paga, uma versão trial (para experimentar) e uma versão acadêmica. Não há mais uma versão gratuita completa do produto (N.R.T.).

www.snow666.blogspot.com
Snow666

servidor em si e suas vulnerabilidades históricas. Ele procura erros de configuração comuns que degradam a segurança e também verifica a presença de bugs conhecidos.

Pelo fato de um relatório gerado por SAINT basicamente fornecer instruções para invadir um sistema, uma pequena, porém sonante, minoria de administradores de sistemas acha que você seria sagaz ao executar SAINT — ou algum programa similar como o Nessus, a seguir — em seus sistemas antes dos hackers.

A interface com o usuário do SAINT baseia-se inteiramente na Web e ela requer que um navegador Web esteja instalado na mesma máquina que o SAINT. Felizmente, o SAINT faz um bom uso do HTML e é capaz de apresentar seus resultados numa série de formatos bem desenhados. SAINT não requer que **nmap** esteja instalado, porém usará **nmap** caso ele se encontre disponível. O SAINT também clama fazer uso dos utilitários fornecidos com o Samba para verificar hosts Windows caso eles tenham sido instalados. Veja www.samba.org ou o Capítulo 27, *Cooperando com o Windows*, para obter maiores informações sobre o Samba.

Nessus: o varredor de redes de próxima geração

Renaud Deraison lançou recentemente um imenso e poderoso pacote chamado Nessus que dispõe praticamente dos mesmos recursos do SAINT, porém de forma mais estruturada e mais fácil de ser expandida. Ele se encontra disponível em www.nessus.org.

O Nessus se orgulha de ser o varredor de segurança que não assume nada como garantido. Em vez de, por exemplo, supor que todos os servidores Web são executados na porta 80, ele fará a varredura em busca de servidores Web sendo executados em qualquer porta e depois os verificará em termos de vulnerabilidades. Em vez de usar o número de versão informado pelo serviço com o qual se conectou, ele tenta explorar vulnerabilidades conhecidas para ver se o serviço é suscetível.

Embora seja necessária uma boa dose de tempo para fazer com que o Nessus fique ativo numa máquina pela primeira vez (ele requer uma série de pacotes que não são instalados com a distribuição-padrão típica), o esforço vale a pena. Na época em que este livro era escrito, o Nessus era capaz de varrer mais de 800 vulnerabilidades comuns através da rede.

O sistema Nessus inclui um cliente e um servidor. O servidor atua como um banco de dados, e o cliente trata da apresentação da interface gráfica com o usuário. Você tem de rodar o servidor num sistema UNIX ou Linux, mas há uma grande variedade de clientes disponíveis para controlar e exibir o Nessus em outras plataformas.

Uma das grandes vantagens do Nessus é o seu desenho modular, que facilita o trabalho de terceiros em adicionar novas verificações de segurança. Se a comunidade de usuários começar a escrever e criar um banco de dados de scripts para o Nessus, ele poderá permanecer atualizado por um longo tempo sem a necessidade de atualizações constantes por parte dos desenvolvedores originais.

crack: localize senhas inseguras

Uma maneira de impedir escolhas de senhas fracas é você mesmo tentar descobrir as senhas e forçar os usuários a alterar as senhas que você foi capaz de descobrir. **crack** é uma ferramenta sofisticada de Alec D. E. Muffett que implementa diversas técnicas para adivinhação de senhas comuns.

Muito embora a grande maioria dos sistemas Linux use um arquivo de senhas-sombra para ocultar senhas criptografadas da visão pública, ainda é inteligente verificar se as senhas de seus usuários são resistentes à ação de **crack**. O conhecimento da senha de um usuário pode ser útil, pois as pessoas tendem a usar a mesma senha repetidamente. Uma única senha pode fornecer acesso a um outro sistema, decifrar arquivos armazenados no diretório de um usuário e permitir acesso a contas financeiras na Web. (Não é preciso dizer que não é muito inteligente em termos de segurança reutilizar uma senha dessa forma. Mas ninguém quer ficar se lembrando de dez senhas.)

Na época em que este livro era escrito, a versão atual do **crack** era 5.0a. Ele pode ser encontrado em ftp.cerias.purdue.edu. Já que a saída de **crack** contém as senhas que ele quebrou, você deve protegê-lo cuidadosamente e eliminá-lo assim que tiver terminado.

tcpd: proteja serviços Internet

tcpd, normalmente conhecido como o pacote de "envoltórios TCP", permite que você registre em log conexões para serviços TCP como **telnetd**, **ftpd** e **fingerd**. Além disso, ele permite que você restrinja quais sistemas podem se conectar a estes serviços. Esses dois recursos podem ser muito práticos quando se está rastreando ou controlando convidados indesejáveis. **tcpd** foi escrito por Wietse Venema; atualizações podem ser encontradas em ftp.porcupine.org. Ele vem como padrão em todas as distribuições Linux.

Snow666

*Consulte a seção 28.3 para obter maiores informações sobre **inetd**.*

tcpd não requer modificações em programas de rede existentes. Ele é montado sobre **inetd**; basta modificar o arquivo **/etc/inetd.conf** para que ele execute **tcpd** em vez do programa de servidor de redes real. Além disso, muitos programas que não são executados por meio de **inetd** (como SSH ou **sendmail**) podem ser compilados para verificar conexões com **tcpd** além de seus próprios arquivos de controle de acesso. **tcpd** realiza então quaisquer verificações de login e de segurança antes de executar o servidor.

Por exemplo, se seu arquivo **/etc/inetd.conf** continha originalmente a linha

```
telnet stream tcp nowait root /usr/sbin/in.telnetd in.telnetd
```

você poderia alterar isso para

```
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
```

Para usuários do Red Hat, a funcionalidade de **tcpd** já vem compilada em **xinetd** e **ssh**.

O arquivo de log resultante (configurado em **/etc/syslog.conf**) ficaria parecido com o seguinte:

```
Nov 12 08:52:43 chimchim in.telnetd[25880]: connect from tintin.Colorado.EDU
Nov 12 19:19:44 chimchim in.telnetd[15520]: connect from catbelly.com
Nov 12 23:48:45 chimchim in.telnetd[19332]: connect from atdt.xor.com
Nov 13 20:14:57 chimchim in.telnetd[2362]: connect from 130.13.13.11
```

COPS: audit system security

COPS (Computer Oracle and Password System) é um conjunto de programas, escrito originalmente por Dan Farmer, que monitora diversos aspectos da segurança no Linux. Trata-se de um pacote bem rudimentar, escrito originalmente em época anterior ao Linux, mas ele ainda é bem útil. Alguns hábitos antigos são difíceis de serem eliminados. O COPS pode ser executado toda noite por meio do **cron** para pesquisar o sistema de arquivos em busca de problemas de segurança.

Padronizando e otimizando uma série de verificações simples, o COPS pode lhe poupar várias horas de trabalho manual. Embora ele não esteja mais em desenvolvimento ativo, o COPS é uma ferramenta clássica que identifica diversos problemas clássicos relacionados com segurança. Execute-o antes de seus usuários o fazerem.

O COPS o alerta sobre problemas em potencial enviando-lhe e-mails; ele não faz nenhuma tentativa de corrigir os problemas que ele descobriu. A seguir, uma lista dos itens monitorados:

- Modos e permissões sobre arquivos, diretórios e dispositivos
- O conteúdo de **/etc/passwd** e **/etc/group**
- O conteúdo dos arquivos crontab e de inicialização de sistema
- A permissão para gravação sobre os diretórios dos usuários

Assim que você instalar o COPS, receberá um relatório de segurança a cada noite similar ao seguinte:

```
ATTENTION:
Security Report from host raja.xor.com

Warning! Root does not own the following file(s): /etc
Warning! "." (or current directory) is in root's path!
Warning! /var/spool/mail is _World_ writable!
Warning! /var/run/utmp is _World_ writable!
Warning! User randy's home directory /home/staff/randy is mode 0777!
Warning! Password file, line 8, no password:
runmailq::33:10:,,,:/home/staff/runmailq:/bin/csh
Warning! Password Problem: Guessed: beth shell: /bin/csh
```

O COPS inclui o sistema especializado Kuang, que tenta intuir maneiras desonestas que os usuários normais poderiam tentar para ganhar acesso root. Mais informações sobre COPS em www.cerias.purdue.edu.

tripwire: monitora alterações em arquivos de sistema

tripwire, criado por Gene Kim e Gene Spafford da Purdue, monitora as permissões e os checksums de importantes arquivos de sistema para que você possa facilmente detectar arquivos que foram substituídos, corrompidos ou adulterados. Por exemplo, **tripwire** facilita determinar se um intruso substituiu sua cópia de **/bin/login** por um que registra senhas num arquivo clandestino.

Snow666

tripwire compara arquivos em relação a um banco de dados que registra suas características e checksums no momento em que o banco de dados foi construído. A idéia geral é criar um banco de dados de referência com base no estado confiável do sistema e depois aplicar **diff** regularmente, comparando o sistema de arquivos com o banco de dados histórico. Arquivos que esperamos que mudem (como **/var/run/utmp**) podem ser marcados no arquivo de configuração de **tripwire** para que não gerem alertas. Quando a configuração do sistema for alterada ou um novo software for instalado, o banco de dados deveria ser reconstruído para que problemas reais não desapareçam no meio de uma enxurrada de alertas **tripwire** espúrios.

Se possível, o arquivo de configuração e o banco de dados de **tripwire** devem ser montados de um servidor seguro que os exporte como arquivos somente para leitura.[11] Essa configuração torna mais difícil para os hackers encobrirem seus rastros e permanecerem despercebidos.

tripwire deve ser configurado para enviar via e-mail um relatório todas as noites. Um relatório típico produzido por **tripwire** seria:

```
# tripwire
Tripwire(tm) ASR (Academic Source Release) 1.3.1
File Integrity Assessment Software
(c) 1992, Purdue Research Foundation, (c) 1997, 1999 Tripwire
Security Systems, Inc. All Rights Reserved. Use Restricted to
Authorized Licensees.
### Phase 1: Reading configuration file
### Phase 2: Generating file list
### Phase 3: Creating file information database
### Phase 4: Searching for inconsistencies
###
###                  Total files scanned: 20344
###                          Files added:     0
###                        Files deleted:     0
###                        Files changed:     1
###
###                 Total file violations:     1
###
changed: -rwxr-xr-x root    262184 Jan 22 12:04:42 2000 /bin/tcsh
### Phase 5:       Generating observed/expected pairs for changed files
###
### Attr        Observed (what it is)     Expected (what it should be)
### ====        =====================     ============================
/bin/tcsh
   st_ctime:    Sat Jan 22 12:04:42 2000 Fri May 14 05:11:41 1999
```

Nesse exemplo, **tripwire** informa que o horário de modificação do inode de **/bin/tcsh** é diferente daquele quando o banco de dados foi gerado. Isso pode ser uma indicação de que um hacker astuto tenha substituído a versão original de **/bin/tcsh** por uma que contenha uma surpresa aguardando para ser descoberta na próxima vez que o shell for executado por root. Comparando-se o checksum do executável com a versão da fita de distribuição (use o utilitário **siggen** que vem com **tripwire** para fazer isto), pode-se confirmar ou negar isso como uma potencial intervenção de hackers. Já que alguns hackers são espertos o bastante para manipular de forma fraudulenta os checksums em arquivos modificados, **tripwire** usa dois métodos de checksum distintos.

tripwire é um tanto incomum já que ele começou como um software gratuito, mas posteriormente foi privatizado e transformado num produto comercial. Porém, não é realmente possível não disponibilizar mais algo que era gratuito anteriormente. A Tripwire, Inc., continuou graciosamente a tornar disponível a versão gratuita e até liberou documentação com qualidade de produto comercial e atualizações para ele. Ele pode ser encontrado (a versão comercial) no site Web deles, www.tripwiresecurity.com, e a versão código-fonte aberto sob a licença GPL no site Web www.tripwire.org.

Ferramentas forenses

Uma poderosa ferramenta de segurança com boas perspectivas de êxito (*kit* de ferramentas, na verdade) é o TCT (The Coroner's Toolkit) de Dan Farmer e Wietse Venema. O TCT é um conjunto de utilitários que ajuda a analisar o sistema após uma invasão de segurança ter ocorrido.

---

11. Manter o banco de dados num disquete ou disco Zip protegidos contra gravação é uma abordagem popular que fornece um nível de segurança moderado.

www.snow666.blogspot.com
Snow666

O TCT ajuda a identificar *o que* aconteceu e *como* isto aconteceu. Em alguns casos, ela irá até recuperar dados que foram destruídos durante a invasão. Um utilitário particularmente interessante é o **mactime**, um programa que rastreia horários de modificação, acesso e de alteração para todos os arquivos do sistema. **mactime** e muitos outros utilitários do kit de ferramentas TCT se encontram disponíveis em www.fish.com/security.

21.9 Ferramentas de Segurança Criptográficas

A maioria dos protocolos de rede usados comumente em sistemas Linux data de uma época anterior ao emprego da Internet e da invenção da criptografia moderna. A segurança simplesmente não era um fator levado em consideração em vários protocolos; em outros, as preocupações com a segurança eram expressas com a transmissão de uma senha em texto puro ou com uma vaga verificação para ver se os pacotes eram provenientes de host ou porta confiável.

Esses protocolos agora se encontram operando nas águas infestadas por tubarões das grandes redes corporativas e da Internet, onde, obrigatoriamente se deve supor que todo o tráfego está aberto à inspeção. Não somente isso, mas há pouco que se possa fazer para impedir que alguém interfira ativamente em conversações que trafegam pelas redes. Como ter certeza de com quem você está realmente falando?

A criptografia fornece uma solução para vários desses problemas. Por um longo tempo foi possível misturar mensagens de forma que alguém na escuta não pudesse decifrá-las, mas isto é apenas o início das maravilhas da criptografia. Desenvolvimentos como criptografia de chaves públicas e hashing seguro permitiram o desenvolvimento de sistemas de criptografia que atendam a praticamente qualquer tipo de necessidade concebível.[12]

Infelizmente, estes avanços matemáticos falharam grandemente em se transformarem em software seguro e utilizável que seja amplamente entendido e adotado. Os desenvolvedores de sistemas de software para criptografia tendem a estar muito interessados em provável exatidão e segurança absoluta e não interessados se um sistema tem um sentido prático para o mundo real. A maioria do software atual tende a ser muito rebuscado em termos de engenharia e talvez não seja surpresa que os usuários saiam correndo quando eles tiverem a mínima chance de fazê-lo. Hoje, a população usando criptografia é formada em grande parte por hobistas interessados no assunto, malucos teóricos acerca de conspiração e aqueles que não têm nenhuma alternativa devido à política administrativa adotada.

Pode ser que vejamos, pode ser que não, uma abordagem mais sã ao desenvolvimento de criptografia ao longo dos próximos anos. Neste meio tempo, algumas ofertas atuais discutidas nas próximas seções poderão ajudá-lo.

Kerberos: uma abordagem unificada à segurança de redes

O sistema Kerberos, desenvolvido no MIT, tenta abordar algumas das questões relativas à segurança das redes de uma maneira consistente e também expansível. Kerberos é um sistema de autenticação, um recurso que "garante" que usuários e serviços sejam realmente quem eles clamam ser. Ele não fornece segurança adicional ou criptografia além dessa.

O Kerberos usa DES para construir conjuntos aninhados de credenciais chamadas "tíquetes". Os tíquetes são repassados pela rede para certificar sua identidade e para lhe dar acesso a serviços de rede. Cada instalação Kerberos deve manter pelo menos uma máquina fisicamente segura (chamada de servidor de autenticação) para executar o daemon Kerberos. Esse daemon emite tíquetes aos usuários ou serviços que solicitam autenticação tomando com base as credenciais que eles apresentam como, por exemplo, senhas.

Em suma, o Kerberos aumenta a segurança de senhas tradicional do Linux em somente duas maneiras: ele jamais transmite senhas decriptografadas pela rede e ele libera os usuários de terem de ficar digitando senhas toda hora, tornando a proteção de senhas de serviços de rede um tanto mais aceitável.

A comunidade Kerberos se orgulha de ter um dos documentos mais lúcidos e agradáveis jamais escrito sobre um sistema criptográfico: "Designing an Authentication System: a Dialogue in Four Scenes", de Bill Bryant. É uma leitura obrigatória para qualquer um que esteja interessado em criptografia e se encontra disponível em

http://web.mit.edu/kerberos/www/dialogue.html

Há também uma excelente FAQ:

http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html

O Kerberos dispõe de um modelo de segurança de rede melhor do que o modelo "ignorar completamente a segurança de uma rede". Entretanto, ele não é perfeitamente seguro nem fácil de ser instalado e utilizado. Ele não substitui nenhuma das medidas de segurança descritas neste capítulo. Em nossa opinião, para a grande parte das ins-

12. Duas excelentes fontes para aqueles interessados em criptografia são "RSA Labs' Frequently Asked Questions about Today's Cryptography" em www.rsasecurity.com/rsalabs/faq e o sci.crypt FAQ disponível via FTP em rtfm.mit.edu.

www.snow666.blogspot.com
Snow666

talações é melhor não optar por ele. Uma boa higiene do sistema e uma solução criptográfica mais focalizada para logins remotos como o SSH (veja a seção SSH: Secure Shell) devem fornecer um nível de segurança mais do que adequado a seus usuários.

Infelizmente, o sistema Kerberos distribuído como parte do Windows 2000 usa extensões proprietárias e não documentadas. Como conseqüência, ele não opera bem com uma distribuição baseada no código do MIT.

PGP: Pretty Good Privacy

O pacote PGP de Philip Zimmermann fornece uma "cesta" de ferramentas com utilitários de criptografia que focalizam essencialmente a segurança de e-mail. Ele pode ser usado para criptografar dados, gerar assinaturas e para verificar a origem de arquivos e mensagens.

Tentativas de regular ou impedir a distribuição do PGP deram a ele uma história bem tumultuada. Hoje ele existe em várias versões, inclusive um conjunto de produtos comerciais da Network Associates (www.nai.com). Uma versão do PGP vetada pelo governo norte-americano se encontra disponível nos Estados Unidos e uma versão internacional com criptografia um tanto mais forte e mais variada se encontra disponível em www.pgpi.org. Os sites de arquivos internacionais parecem não impedir o acesso aos endereços dos Estados Unidos, de modo que usuários norte-americanos devem tomar cuidado para não irem acidentalmente a www.pgpi.org e fazer o download da versão completa do PGP.

O PGP é o software para criptografia mais popular de uso comum. Infelizmente, a versão UNIX/Linux é suficientemente complexa a ponto de exigir que você entenda um bocado de conceitos de criptografia para poder usá-lo. Felizmente (?), o PGP vem com um tratado de 88 páginas sobre criptografia que pode ajudá-lo a preparar o terreno. Embora você possa achar o PGP útil em seu próprio trabalho, nós não recomendamos que dê suporte para os usuários, já que é sabido que ele gera muitas perguntas enigmáticas. Descobrimos que a versão para Windows do PGP é consideravelmente mais fácil de ser usada do que o comando **pgp** do Linux com seus 38 modos diversos de operação.

Os pacotes de software na Internet são normalmente distribuídos com um arquivo de assinatura PGP que garante a origem e a pureza do software. Infelizmente, é difícil para as pessoas que não são usuários obstinados do PGP validarem essas assinaturas – não pelo processo de validação ser complicado, mas porque a verdadeira segurança PGP pode provir somente da coleta de uma biblioteca pessoal de chaves públicas de pessoas cujas identidades você verificou diretamente. Fazer o download de uma única chave pública juntamente com um arquivo de assinatura e distribuição de software é aproximadamente tão seguro quanto fazer o download da distribuição sozinha.

Infelizmente, há algumas questões de patentes associadas à tecnologia em que o PGP se baseia. O GnuPG — GNU Privacy Guard, www.gnupg.org — (também conhecido como GPG) é uma alternativa ao PGP que fornece a mesma funcionalidade básica sem usar o algoritmo IDEA patenteado. Ele mantém um nível razoável de compatibilidade com implementações modernas do PGP e se encontra prontamente disponível para Linux. Há atualmente alguns programas que apresentam interfaces gráficas amigáveis para os usuários, o que facilita o uso de sistemas do tipo PGP ou GPG.

SSH: Secure SHell

O sistema SSH, criado por Tatu Ylönen, é um substituto seguro para **rlogin**, **rcp** e **telnet**. Ele usa autenticação criptográfica para confirmar a identidade de um usuário e criptografa todo o fluxo de comunicações entre os dois hosts. O protocolo usado pelo SSH é concebido para resistir a uma série de ataques em potencial. Ele foi bem estudado e está sendo padronizado pelo IETF.

Assim como o **tripwire**, o SSH sofreu uma transformação, passando de um projeto de fonte aberto distribuído gratuitamente (SSH1) para se tornar um produto comercial usando um protocolo ligeiramente diferente (e mais seguro) (SSH2). Felizmente, a comunidade dos fontes abertos respondeu lançando o OpenSSH, que agora implementa ambos os protocolos.

É importante notar que uma série de vulnerabilidades foram descobertas no SSH original, no SSH ora gratuito e depois comercial e também no OpenSSH. Conseqüentemente, é muito importante verificar que você esteja executando a versão mais recente e "mais segura". Um artigo publicado na USENIX em dezembro de 2001 incluía estatísticas demonstrando que mais de 30% das instalações usando SSH atualmente estão usando uma versão vulnerável. Geralmente, recomendamos a versão OpenSSH, disponível em www.openssh.com.

Os principais componentes do SSH são um daemon de servidor, **sshd**, e dois comandos em nível de usuário: **ssh** para logins remotos e **scp** para copiar arquivos. Outros componentes são um comando **ssh-keygen** que gera pares de chaves públicas e uma série de utilitários que auxiliam no suporte a X Windows seguro.

sshd pode autenticar logins de usuários de diversas maneiras diferentes. Fica a seu cargo como administrador decidir quais destes métodos são aceitáveis:

Snow666

- **Método A:** Se o nome do host remoto por meio do qual o usuário está fazendo o login estiver listado em **~/.rhosts**, **~/.shosts**, **/etc/hosts.equiv** ou **/etc/shosts.equiv**, então o usuário entra no sistema automaticamente sem necessidade de verificação de uma senha. Este esquema espelha aquele do antigo daemon **rlogin** e, em nossa opinião, *não* é aceitável para emprego comum.
- **Método B:** Como um refinamento do método A, **sshd** também pode usar criptografia de chaves públicas para verificar a identidade do host remoto. Para isso acontecer, a chave pública do host remoto (gerado no momento da instalação) tem de ser listada no arquivo **/etc/ssh_known_hosts** do host local ou no arquivo **~/.ssh/known_hosts** do usuário. Se o host remoto puder provar que ele conhece a chave privada correspondente (normalmente armazenada em **/etc/ssh_host_key**, um arquivo que não pode ser lido por todos), então o usuário entra no sistema sem lhe ser solicitado uma senha. O método B é mais restritivo do que o método A, porém acreditamos que ele ainda não seja tão seguro. Se a segurança do host de origem estiver comprometida, a instalação local também estará comprometida.
- **Método C: sshd** pode usar criptografia de chaves públicas para estabelecer a identidade do usuário. No momento do login, o usuário precisa ter acesso a uma cópia de seu arquivo de chaves públicas e tem de fornecer uma senha para decifrá-la. Este método é o mais seguro, mas é um tanto chato de ser configurado. Isso também significa que você não pode entrar no sistema quando estiver fora a menos que leve uma cópia de seu arquivo de chaves privadas consigo (talvez em seu laptop).
- **Método D:** Por fim, **sshd** pode simplesmente permitir que o usuário introduza sua senha de login normal. Isso faz com que **ssh** se comporte de forma muito semelhante a **telnet**, exceto pelo fato de a senha e sessão serem ambas criptografadas. Os principais inconvenientes deste método são o fato de as senhas de login de sistema serem relativamente fracas (em geral limitadas a 8 caracteres significativos) e pelo fato de haver ferramentas prontas (como **crack**) desenhadas para quebrá-las. Entretanto, este método provavelmente seja a melhor opção para uso normal.

A política de autenticação é configurada no arquivo **/etc/sshd_config**. Você verá logo que esse arquivo foi preenchido com lixo de configuração, mas grande parte dele pode ser ignorado sem problemas. As opções relevantes à autenticação são mostradas na Tabela 21.1.

Tabela 21.1 Opções em /etc/sshd_config relacionadas com a autenticação.

Opção	Método[a]	Padrão	Significado quando ativa
RhostsAuthentication	A	não	Permite login via **~/.shosts**, **/etc/shosts.equiv** etc.
RhostsRSAAuthentication	B	sim[d]	Permite **~/.shosts** et al., mas também requer chave de host
IgnoreRhosts	A,B	não	Ignora os arquivos **~/.rhosts** e **hosts.equiv**[b]
IgnoreRootRhosts	A,B	não[c]	Impede a autenticação **rhosts/shosts** para root
RSAAuthentication	C	sim	Permite autenticação com criptografia de chaves públicas por usuário
PasswordAuthentication	D	sim	Permite o emprego de senha de login normal

a. Os métodos de autenticação para os quais esta variável é relevante.
b. Mas continua a honrar **~/.shosts** e **shosts.equiv**.
c. O padrão é o valor de IgnoreRhosts; esta opção não está presente na versão 3.6 e seguintes.
d. Na versão 3.6 e na atual (3.7) do OpenSSH o padrão é "não".

Nossa configuração sugerida, que permite os métodos C e D, mas não os métodos A ou B, é mostrada a seguir.

```
RhostsAuthentication no
RhostsRSAAuthentication no
RSAAuthentication yes
PasswordAuthentication yes
```

OPIE: One-time Passwords In Everything

Um dos problemas com sistemas como o SSH é que ambas as extremidades de uma conexão têm de suportar um protocolo especial para assegurar a conexão. Isso normalmente não é um problema, porém os usuários podem algumas vezes se sentir encalhados. Clientes SSH não estão disponíveis para todos os sistemas operacionais (pelo menos, gratuitamente não) e os usuários podem ocasionalmente querer entrar no sistema através de computadores de terceiros quando estiverem longe de seu escritório.

Snow666

O padrão OTP (One-Time Password) definido na RFC1938 adota uma abordagem um tanto diferente em termos de segurança de senhas: em vez de criptografar senhas, você somente se certifica de que elas funcionem uma só vez. Senhas em forma de texto puro podem então ser digitadas impunemente através da rede, já que não faz diferença se alguém as escutar. Normalmente os usuários imprimem uma série de senhas usadas somente uma vez para levá-las consigo. Diferentemente de senhas comuns, as senhas usadas uma só vez são geradas em seu nome, você não tem de selecioná-las.

OPIE é o sistema OTP mais comumente usado hoje. Ele é uma ramificação de um sistema anterior chamado S/Key da Bellcore (hoje, Telcordia Technologies) que foi mais desenvolvido no U.S. Naval Research Laboratories. Os principais recursos do OPIE são versões compatíveis com OTP de **telnetd** e **ftpd** e utilitários para geração e administração de listas de senhas. Ele se encontra disponível em www.inner.net/pub/opie.

É importante notar que sistemas OTP tratam somente da questão de bisbilhotice de senhas. Eles não são capazes e não criptografam o real conteúdo de uma conversação. Alguém escutando em sua sessão **telnet** talvez não esteja apto a obter uma senha utilizável, porém ele certamente poderia descobrir muitas coisas a respeito de sua conta. Qualquer senha que você digitasse após o login (para **sudo**, por exemplo) estaria completamente exposta.[13]

Dado à crescente disponibilidade de sistemas como SSH, há cada vez menos necessidade do OPIE. Caso realmente não precise dele, não o utilize – ele é problemático para ser instalado e mantido, e os procedimentos que os usuários devem seguir são um tanto confusos.

Identificadores de hardware

Atualmente, as aparições mais comuns de senhas para serem usadas somente uma vez (OTP) estão em produtos comerciais para segurança. Uma série de fabricantes oferece sistemas com senhas para serem usadas somente uma vez baseados em pequenos dispositivos do tamanho de um "chaveiro" (ou do tamanho de um cartão de crédito) com displays LCD que geram senhas no ato para logins de usuário. No mínimo, distribuir e substituir pequenos dispositivos de hardware manterá o seu assistente administrativo ocupado por algumas horas por semana.

21.10 FIREWALLS

Além de proteger máquinas individuais, você também pode implementar algumas precauções de segurança em nível de rede. A ferramenta básica de segurança de rede é o "firewall". Há três categorias principais de firewalls: filtragem de pacotes, proxy[14] de serviços e inspeção de estados.

Firewalls para filtragem de pacotes

Um firewall para filtragem de pacotes limita os tipos de tráfego que podem passar através de seu gateway Internet (ou através de um gateway interno que separa domínios dentro de sua empresa) baseado em informações contidas no cabeçalho do pacote. É bem semelhante atravessar uma fronteira internacional de carro passando por um posto alfandegário. Você especifica quais são os endereços de destino, números de portas e tipos de protocolos que são aceitáveis, e o gateway simplesmente descarta (e em alguns casos, registra em logs) pacotes que não atendam ao perfil preestabelecido.

A filtragem de pacotes é suportada por roteadores dedicados como aqueles fabricados pela Cisco. Eles também podem estar disponíveis em software, dependendo da máquina que você estiver usando como gateway e de sua configuração. Em geral, os firewalls para filtragem de pacotes oferecem um aumento significativo na segurança com baixo custo em termos de desempenho ou de complexidade.

O Linux inclui software para filtragem de pacotes (veja os detalhes na seção 21.11 para maiores informações). Também é possível comprar software comercial para desempenhar essa função. Todos esses pacotes possuem valores especiais podendo fornecer um firewall razoavelmente seguro para ambientes domésticos ou de pequenos escritórios. Entretanto, você deve se referir aos comentários no início deste capítulo antes de considerar um sistema Linux como um firewall corporativo em nível de produção.[15] Esse é um caso no qual você realmente deve gastar dinheiro num aplicativo de rede dedicado como, por exemplo, o firewall PIX da Cisco.

---

13. Em particular, os usuários não devem tentar obter uma lista de mais senhas OTP depois de terem entrado no sistema com OTP; todas as senhas serão transmitidas sem proteção.
14. O termo "proxy" é um tanto comum no jargão de informática. Estritamente, ele significa procurador, alguém que atua em nome de outro. (N.R.T.)
15. Nós partimos do pressuposto que você já saiba que não deve levar em consideração algo como o Windows como plataforma de firewall. O nome "Windows" evoca imagens de segurança? Bobinho, o Windows é destinado a desktops.

Snow666

Como os serviços são filtrados

A maioria dos serviços bem conhecidos está associada a uma porta de rede no arquivo **/etc/services** ou seu equivalente específico a um determinado fabricante. Os daemons que fornecem estes serviços vinculam-se às portas apropriadas e ficam aguardando conexões de sites remotos[16]. A maioria das portas de serviços bem conhecidas são "privilegiadas", significando que seus números de portas se encontram dentro do intervalo de 1 a 1.023. Essas portas podem ser usadas somente por um processo que esteja sendo executado como root. As portas de números acima de 1.024 são conhecidas como portas não privilegiadas.

A filtragem de serviços específicos se baseia na suposição de que o cliente (a máquina que inicia uma conversação TCP ou UDP) usará uma porta não privilegiada para contactar uma porta privilegiada no servidor. Por exemplo, se você quisesse permitir somente conexões SMTP que chegassem para uma máquina com o endereço 192.108.21.200, você instalaria um filtro que permitisse pacotes TCP destinados àquele endereço na porta 25 e que permitisse pacotes TCP que saíssem daquele endereço para qualquer lugar.[17] A maneira exata como tal filtro seria instalado dependeria do tipo de roteador que você estivesse usando.

*Consulte a seção 22.6 para obter maiores informações sobre como configurar um servidor **ftp**.*

Alguns serviços, como o FTP, acrescentam uma variável inesperada ao quebra-cabeças. O protocolo FTP na verdade usa duas conexões TCP ao transferir um arquivo: uma para comandos e a outra para dados. O cliente inicia a conexão para comandos e o servidor inicia a conexão para dados. Portanto, se você quiser usar FTP para recuperar arquivos da Internet, terá de permitir acesso de entrada a todas as portas TCP não privilegiadas, já que você não tem a mínima idéia de qual porta poderia ser usada para formar uma conexão de dados de entrada.

Esse ajuste frustra em grande parte o objetivo da filtragem de pacotes, pois alguns serviços notoriamente inseguros (por exemplo, X11 na porta 6000) naturalmente se vinculam a portas não privilegiadas. Essa configuração também cria uma oportunidade para usuários curiosos dentro de sua empresa iniciarem seus próprios serviços (como um servidor **telnet** numa porta não padronizada e não privilegiada) que eles ou seus amigos poderão acessar pela Internet.

A maneira mais segura de usar um filtro de pacotes é iniciar com uma configuração que não permita nada a não ser SMTP que chega. A seguir você pode liberar o filtro pouco a pouco à medida que for descobrindo coisas úteis que não funcionam.

Algumas instalações extremamente conscientizadas dos problemas referentes à segurança usam filtragem de dois estágios. Nesse esquema, um filtro é um gateway para a Internet e um segundo filtro se localiza entre o gateway externo e o restante da rede local. A idéia é deixar o gateway externo relativamente aberto e configurar o gateway interno de maneira conservadora. Se a máquina intermediaria for administrativamente separada do restante da rede, ela poderá fornecer uma variedade de serviços na Internet com risco reduzido.

Uma abordagem razoável ao dilema FTP é permitir FTP para o mundo externo somente por meio deste único host isolado. Os usuários poderão fazer login na máquina FTP quando precisarem realizar outras operações de rede que são proibidas pela rede interna. Já que replicar todas as contas de usuário no "servidor" FTP frustraria o objetivo da separação administrativa, talvez seja interessante criar contas FTP somente por solicitação. Naturalmente, o host FTP deveria rodar um conjunto completo de ferramentas para verificação da segurança do sistema.

Firewalls proxy de serviços

Os proxies de serviço interceptam conexões de e para o mundo externo e estabelecem novas conexões a serviços dentro de sua rede, atuando como uma espécie de serviço de leva e traz ou "acompanhante vigilante" entre os dois mundos. É bem parecido com ir de carro até a fronteira de seu país, atravessar a fronteira a pé e alugar um carro limpinho do outro lado da fronteira para continuar a viagem.

Devido ao seu desenho, os firewalls proxy de serviços são muito menos flexíveis (e muito mais lentos) do que filtros de pacotes puros. Seu proxy deve ter um módulo que decodifique e transporte cada protocolo que você quiser que passe pelo firewall. No início dos anos 90 isso era relativamente fácil, pois havia somente alguns protocolos usados comumente. Hoje em dia, os internautas podem utilizar várias dezenas de protocolos em uma hora de navegação na rede. Como resultado, os proxies de serviços são relativamente impopulares em empresas que usam a Internet como meio primário para comunicação.

16. Em muitos casos **inetd** realiza a espera real em nome desses servidores. Veja a seção 28.3 para maiores informações.
17. A porta 25 é a porta SMTP conforme definido em **/etc/services**.

www.snow666.blogspot.com
Snow666

Firewalls de inspeção com estados

A teoria por trás dos firewalls de inspeção com estados está no fato de que se você pudesse escutar cuidadosamente e compreender todas as conversações (em todas as linguagens) que estivessem ocorrendo num aeroporto lotado, você poderia ter certeza de que ninguém estaria planejando colocar uma bomba num avião mais tarde naquele dia. Os firewalls de inspeção com estados são desenhados para inspecionar o tráfego que os atravessa e comparar a real atividade de rede com aquela que teoricamente "deveria" estar acontecendo. Por exemplo, se os pacotes trocados numa seqüência de comandos FTP citarem uma porta para ser usada posteriormente para uma conexão de dados, o firewall deve esperar que uma conexão de dados ocorra somente naquela porta anunciada. Tentativas do site remoto de se conectar a outras portas provavelmente são adulteradas e devem ser eliminadas.

Infelizmente, a realidade normalmente mata o gato aqui mesmo. Não é mais realístico rastrear o "estado" das conexões de rede de milhares de hosts usando centenas de protocolos do que ouvir cada conversa em cada uma das línguas num aeroporto internacional lotado. Algum dia, quando a capacidade de processamento e de memória aumentarem, talvez isso seja factível.

Portanto, o que realmente os fabricantes estão vendendo quando eles dizem fornecer inspeção com estados? Os seus produtos ou monitoram um número muito limitado de conexões ou protocolos, ou então eles tentam localizar um dado conjunto de situações "ruins". Não que haja alguma coisa de errada nisso; certamente algum benefício se obtém de qualquer tecnologia capaz de detectar anomalias de tráfego. Nesse caso em particular, entretanto, é importante se lembrar de que as alegações são, *em grande parte*, apenas exageros de marketing.

Firewalls: qual o verdadeiro nível de segurança oferecido por eles?

Um firewall não deve ser seu meio de defesa primário contra a ação de intrusos. Ele somente é apropriado como uma medida de segurança complementar. O emprego de firewalls normalmente fornece um falso sentido de segurança. Se ele o induzir a relaxar em outras medidas de segurança, terá um efeito *negativo* na segurança de sua instalação.

Todo host dentro de sua organização deve ser assegurado individualmente e monitorado regularmente com ferramentas como **crack**, **tcpd**, **nmap**, COPS e **tripwire**. Da mesma maneira, toda a sua comunidade de usuários precisa ser instruída sobre a higiene básica em termos de segurança. Caso contrário, você estará simplesmente erigindo uma estrutura que tem uma casca dura e um miolo macio. Na Internet, não é preciso muito para se chegar a esse miolo tipo bombom.

De forma ideal, os usuários locais devem ser capazes de se conectar a qualquer serviço que queiram, porém as máquinas na Internet devem ser capazes de se conectar somente a um conjunto limitado de serviços locais. Por exemplo, talvez você queira permitir acesso FTP a um servidor de arquivos local e permitir conexões SMTP (e-mail) para o seu servidor de e-mail.

Caso queira maximizar o valor de sua conexão Internet, recomendamos que você enfatize a conveniência e a acessibilidade ao decidir como configurar uma rede. No final do dia, é a vigilância do administrador de sistemas que torna uma rede segura, não um hardware de firewall estrambótico.

21.11 Recursos de Firewall no Linux: Tabelas IP

Normalmente não recomendamos o emprego de sistemas Linux (ou UNIX ou NT) como firewalls devido à insegurança geral de executar um sistema operacional completo. Dispositivos embutidos desenhados especificamente para roteamento e filtragem de pacotes (como um computador PIX da Cisco) formam os melhores firewalls. Entretanto, um firewall Linux é melhor do que nenhum firewall para uma instalação doméstica ou uma que não possua os recursos financeiros necessários para comprar o hardware apropriado, de modo que descreveremos nesta seção o software Linux para filtragem de pacotes.

Se você estiver decidido a usar uma máquina Linux como firewall, pelo menos se certifique de que ela esteja atualizada em relação a correções e configurações relativas à segurança de sistemas. Uma máquina firewall é um excelente local para colocar em prática todas as recomendações deste capítulo. (A seção 21.10 discute os firewalls para filtragem de pacotes em geral. Caso não esteja familiarizado com o conceito básico de firewall, seria interessante ler essa seção antes de prosseguir.)

A versão 2.4 do kernel Linux introduziu um mecanismo de manipulação de pacotes todo novo chamado Netfilter. A ferramenta usada para controlar Netfilter, **iptables**, é o irmão mais velho do antigo comando **ipchains** usado nos kernels 2.2 do Linux. **iptables** aplica "cadeias" ordenadas de regras a pacotes de rede. Conjuntos de cadeias formam as "tabelas" e estas são usadas para manipular tipos de tráfego específicos.

Por exemplo, a tabela **iptables** padrão se chama "filtro". Cadeias de regras nessa tabela são usadas para filtragem dos pacotes do tráfego de rede. A tabela de filtros contém três cadeias-padrão. Cada pacote que é manipulado pelo kernel passa exatamente através de uma dessas cadeias. As regras na cadeia FORWARD são aplicadas a todos

Snow666

os pacotes que chegam em uma interface de rede e precisam ser encaminhados a uma outra. As regras contidas nas cadeias INPUT e OUTPUT são aplicadas, respectivamente, a tráfego destinado a ou originando-se do host local. Essas três cadeias-padrão normalmente são tudo o que você precisa para aplicar um firewall entre duas interfaces de rede. Se necessário, você pode definir uma configuração personalizada para suportar situações de roteamento ou de contabilidade mais complexas.

Além da tabela de filtros, **iptables** inclui as tabelas "nat" e "mangle". A tabela nat contém cadeias de regras que controlam Network Address Translation (neste caso, "nat" é o nome da tabela **iptables** e "NAT" é o nome do esquema genérico para conversão de endereços). O tópico *Endereços privados e o NAT* na seção 13.4 discute o NAT, e um exemplo da tabela nat em ação é mostrado na seção 13.12. Posteriormente nesta seção, usamos a cadeia PREROUTING da tabela nat para filtragem de pacotes antibisbilhotice.

A tabela mangle contém cadeias que modificam ou alteram o conteúdo dos pacotes de rede fora do contexto da filtragem de pacotes e NAT. Embora a tabela mangle seja útil para manipulação de pacotes especiais como, por exemplo, reinicializar o tempo de vida (TTL) do cabeçalho IP, ela não é normalmente usada na maioria dos ambientes de produção. Nesta seção, discutimos somente as tabelas de filtragem e nat, deixando a tabela mangle para os aventureiros.

Cada regra que forma uma cadeia possui uma estrutura "alvo" que determina o que fazer com os pacotes que atendem às regras. Assim que um pacote atende a uma determinada regra, seu destino é selado e nenhuma regra adicional é verificada. Embora muitos destinos sejam definidos internamente às **iptables**, é possível especificar uma outra cadeia como destino de uma regra.

Os alvos disponíveis a essas regras na tabela de filtragem são ACCEPT, DROP, REJECT, LOG, MIRROR, QUEUE, REDIRECT, RETURN e ULOG. Quando uma regra resultar num ACCEPT, os pacotes que atendem a essa regra terão permissão para prosseguirem em sua trajetória. DROP e REJECT descartam seus pacotes. DROP é silencioso e REJECT retorna uma mensagem de erro ICMP. LOG fornece uma maneira simples de rastrear pacotes à medida que atendem a regras, e ULOG fornece registro de log estendido.

REDIRECT desvia pacotes para um proxy em vez de deixá-los passar em suas trajetórias felizes. Você poderia usar esse recurso para forçar que todo o tráfego Web de seu site passasse através de um cache Web como o Squid. RETURN termina cadeias definidas por usuários e é análogo à instrução return na chamada de uma sub-rotina. MIRROR permuta os endereços IP de origem e de destino antes de enviar o pacote. Finalmente, QUEUE encaminha pacotes a programas de usuários locais por meio de um módulo de kernel.

Um firewall Linux é implementado normalmente como uma série de comandos **iptables** contidos num script de inicialização **rc**. Comandos **iptables** individuais normalmente assumem uma das seguintes formas:

```
iptables -F chain-name
iptables -P chain-name target
iptables -A chain-name -i interface -j target
```

A primeira forma (**-F**) limpa regras anteriores da cadeia. A segunda forma (**-P**) configura uma política-padrão (também conhecida como destino) para a cadeia. Recomendamos que você use DROP como destino para a cadeia-padrão. A terceira instância (**-A**) anexa a especificação atual à cadeia. A menos que você especifique uma tabela com o argumento **-t**, seus comandos se aplicarão a cadeias na tabela de filtragem. O parâmetro **-i** aplica a regra à interface designada e **-j** identifica o alvo. **iptables** aceita muitas outras estruturas, algumas das quais são mostradas na Tabela 21.2.

Tabela 21.2 Opções de linha de comando para filtros iptables.

Estrutura	Significado ou valores possíveis
-p *proto*	Concordâncias por protocolo: **tcp**, **udp** ou **icmp**
-s *source-ip*	Concorda com o endereço IP de origem de rede ou host (a notação CIDR é OK)
-d *dest-ip*	Concorda com o endereço de destino de rede ou host
--sport *port#*	Concorda com a porta de origem (observe os dois traços)
--dport *port#*	Concorda com a porta de destino (observe os dois traços)
--icmp-type *type*	Concorda com o código de tipo ICMP (observe os dois traços)
!	Nega uma estrutura
-t *table*	Especifica a tabela à qual um comando se aplica (o padrão é **filter**)

Snow666

A seguir analisamos um exemplo completo. Partimos do pressuposto que a interface ppp0 vai para a Internet e que a interface eth0 vai para uma rede interna. O endereço IP de ppp0 é 128.138.101.4, o endereço IP de eth0 é 10.1.1.254 e ambas as interfaces possuem uma máscara de rede equivalente a 255.255.255.0. Este exemplo usa filtragem de pacotes sem estado para proteger o servidor Web com o endereço IP 10.1.1.2, que é o método-padrão para proteger servidores Internet. Posteriormente no exemplo, mostraremos como usar filtragem com estados para proteger usuários desktop.

Antes de poder usar **iptables**, é necessário habilitar o encaminhamento IP e certificar-se de que os diversos módulos **iptables** tenham sido carregados no kernel. Para maiores informações sobre como habilitar encaminhamento IP, veja *Ajustando um kernel Linux* na seção 12.4 ou *Variáveis de kernel relacionadas com segurança* na seção 13.11. Todas as distribuições que vêm com **iptables** também vêm com scripts de inicialização para se conseguir fazer essa habilitação e carga.

Nosso primeiro conjunto de regras inicializa a tabela de filtros. Inicialmente, todas as cadeias são esvaziadas, depois o destino-padrão das cadeias INPUT e FORWARD são configurados em DROP. Como acontece com qualquer outro firewall de rede, a estratégia mais segura é a de descartar quaisquer pacotes que você não tenha dado permissão explícita.

```
iptables -F
iptables -P INPUT DROP
iptables -P FORWARD DROP
```

Já que as regras são avaliadas na ordem em que aparecem numa cadeia, colocaremos nossas regras mais atarefadas na frente.[18] As três primeiras regras na cadeia FORWARD permitem conexões através do firewall a serviços de rede em 10.1.1.2. Mais especificamente, permitimos SSH (porta 22), HTTP (porta 80) e HTTPS (porta 443) por meio de nosso servidor Web. A primeira regra permite todas as conexões através do firewall originárias do interior da rede confiável.

```
iptables -A FORWARD -i eth0 -p ANY -j ACCEPT
iptables -A FORWARD -d 10.1.1.2 -p tcp --dport 22 -j ACCEPT
iptables -A FORWARD -d 10.1.1.2 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -d 10.1.1.2 -p tcp --dport 443 -j ACCEPT
```

O único tráfego TCP que permitiremos para o nosso host firewall será SSH, que será útil para gerenciamento do firewall. A segunda regra listada a seguir permite tráfego loopback, que permanece local para nosso host de firewall. Nossos administradores ficam nervosos quando não conseguem efetuar um **ping** em suas rotas-padrão, de forma que a terceira regra aqui permite pacotes ICMP ECHO_REQUEST por meio de nossos endereços IP internos.

```
iptables -A INPUT -i eth0 -d 10.1.1.1 -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -i lo -d 127.0.0.1 -p ANY -j ACCEPT
iptables -A INPUT -i eth0 -d 10.1.1.1 -p icmp --icmp-type 8 -j ACCEPT
```

Para qualquer host TCP/IP funcionar adequadamente na Internet, determinados tipos de pacotes ICMP têm de ter permissão para passar pelo firewall. As oito regras a seguir permitem um conjunto mínimo de pacotes ICMP no host para firewall, bem como na rede que está por trás dele.

```
iptables -A INPUT -p icmp --icmp-type 0 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 3 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 5 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 11 -j ACCEPT
iptables -A FORWARD -d 10.1.1.2 -p icmp --icmp-type 0 -j ACCEPT
iptables -A FORWARD -d 10.1.1.2 -p icmp --icmp-type 3 -j ACCEPT
iptables -A FORWARD -d 10.1.1.2 -p icmp --icmp-type 5 -j ACCEPT
iptables -A FORWARD -d 10.1.1.2 -p icmp --icmp-type 11 -j ACCEPT
```

A seguir adicionamos regras à cadeia PREROUTING na tabela nat. Embora a tabela nat não se destine à filtragem de pacotes, sua cadeia PREROUTING é particularmente útil para filtragem antibisbilhotice. Se colocarmos entradas DROP na cadeia PREROUTING, elas não precisarão estar presentes nas cadeias INPUT e FORWARD, já que a cadeia PREROUTING é aplicada a todos os pacotes que entram no host firewall. É mais elegante colocar as entradas num único lugar em vez de duplicá-las.

```
iptables -t nat -A PREROUTING -i ppp0 -s 10.0.0.0/8 -j DROP
```

18. Porém, você deve estar ciente de que reordenar regras para fins de desempenho não altera a funcionalidade.

Snow666


```
iptables -t nat -A PREROUTING -i ppp0 -s 172.16.0.0/12 -j DROP
iptables -t nat -A PREROUTING -i ppp0 -s 192.168.0.0/16 -j DROP
iptables -t nat -A PREROUTING -i ppp0 -s 127.0.0.0/8 -j DROP
iptables -t nat -A PREROUTING -i ppp0 -s 224.0.0.0/4 -j DROP
```

Finalmente, encerramos as cadeias INPUT e FORWARD com uma regra que proíbe todos os pacotes que não possuem permissão explícita. Embora já tenhamos forçado esse comportamento através dos comandos **iptables -P**, o alvo LOG nos permite ver quem da Internet está batendo à nossa porta.

```
iptables -A INPUT -i ppp0 -j LOG
iptables -A FORWARD -i ppp0 -j LOG
```

Opcionalmente, poderíamos configurar IP NAT para dissimular o espaço de endereços privados usado na rede interna. Veja *NAT no Linux (mascaramento IP)* na seção 13.12 para obter maiores informações sobre o NAT.

Um dos recursos mais poderosos que Netfilter traz para os firewalls Linux é a filtragem de pacotes com estados. Em vez de permitir a entrada de serviços específicos, um firewall para clientes conectado à Internet precisa permitir respostas que chegam às solicitações dos clientes. A cadeia FORWARD com estados simples a seguir permite que todo o tráfego deixe nossa rede, porém autoriza somente tráfego que chega que esteja relacionado às conexões iniciadas pelos nossos hosts.

```
iptables -A FORWARD -i eth0 -p ANY -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Certos módulos de kernel têm de ser carregados para habilitar **iptables** a rastrear sessões de rede complexas como aquelas de FTP e IRC. Se esses módulos não estiverem carregados, **iptables** simplesmente não autoriza tais conexões. Embora filtros de pacotes de estados possam aumentar a segurança de sua instalação, eles também adicionam complexidade à rede. Certifique-se de que você realmente precisa desse recurso de estados antes de implementá-lo em seu firewall.

Talvez a maneira mais fácil de depurar conjuntos de regras **iptables** é usar **iptables -L -v**. Essas opções lhe informarão quantas vezes cada regra em suas cadeias coincidiram com um pacote. Normalmente adicionamos regras **iptables** temporárias com o alvo LOG quando queremos informações adicionais sobre os pacotes que coincidiram com as regras preestabelecidas. Muitas vezes se pode resolver problemas mais complicados usando-se um pacote analisador como o **tcpdump**.

21.12 VPNs (Virtual Private Networks)

Um dos desenvolvimentos mais interessantes nos últimos anos foi o advento das VPNs (Virtual Private Networks, redes privadas virtuais). Esta tecnologia se tornou possível em grande parte pelo aumento do poder de processamento que agora se obtém de um único chip (nas estações de trabalho dos usuários). Em sua forma mais simples, uma VPN é uma conexão que faz com que uma rede remota pareça estar conectada diretamente, mesmo que ela esteja fisicamente a milhares de quilômetros e a muitos saltos de roteador de distância. Para maior segurança, a conexão não é somente autenticada de alguma maneira (normalmente através de um "segredo compartilhado" como uma senha), como também o tráfego de ponta a ponta é também criptografado. Tal arranjo é normalmente conhecido como "túnel seguro".

Eis um bom exemplo do tipo de situação na qual uma VPN é útil: suponha que uma empresa tenha filiais em Chicago, Boulder e Miami. Se cada filial tiver uma conexão com um provedor de serviços Internet local, a empresa poderá usar VPNs para conectar os escritórios de modo transparente (e, em grande parte, de forma segura) através da Internet não segura. A empresa poderia atingir um resultado similar alugando linhas dedicadas para conectar três escritórios, mas esta opção poderia ser consideravelmente mais cara.

Um outro bom exemplo poderia ser uma empresa cujos funcionários fazem telecomutação de suas residências. As VPNs permitiriam que esses usuários colhessem os frutos de seus serviços de modem a cabo baratos e de alta velocidade enquanto ainda fariam com que eles parecessem diretamente conectados à rede corporativa.

Devido à conveniência e à popularidade desta funcionalidade, todo o mundo está oferecendo algum tipo de solução VPN. Você pode adquiri-la de seu fabricante de roteador, na forma de um plug-in para o seu sistema operacional ou até mesmo na forma de um dispositivo VPN dedicado para sua rede. Dependendo de seu orçamento e de suas necessidades de expansão, talvez seja interessante considerar umas das várias soluções VPN comerciais existentes no mercado.

Caso esteja com o orçamento curto e querendo uma solução rápida, o SSH fará o tunelamento seguro para você. O SSH é usado normalmente para fornecer uma conectividade uma porta por vez, mas ele também pode fornecer funcionalidade pseudo-VPN como ilustrado em um exemplo mais à frente, nesta seção, a que usa PPP através de um túnel SSH.

Snow666

Túneis IPSEC

Caso seja um fã dos padrões IETF (ou de economizar dinheiro) e precise de uma verdadeira solução VPN, você deverá dar uma olhada no IPSEC (Internet Protocol SECurity). O IPSEC foi originalmente desenvolvido para IPv6, mas ele também foi largamente implementado para IPv4. O IPSEC é um sistema de criptografia e de autenticação ponta-a-ponta aprovado pelo IETF. Quase todos os fornecedores VPN sérios oferecem um produto que tenha ao menos um modo de compatibilidade IPSEC.

O IPSEC usa criptografia forte para fornecer serviços de autenticação, bem como de criptografia. A autenticação garante que os pacotes sejam provenientes do emitente correto e, não foram alterados em trânsito, e a criptografia impede o exame desautorizado do conteúdo dos pacotes. Infelizmente, o emprego do IPSEC tem sido um tanto dificultado pelas leis de criptografia dos Estados Unidos, que proíbe a exportação de software para criptografia forte.

Em sua forma atual, o IPSEC criptografa o cabeçalho da camada de transporte, incluindo os números de portas de origem e de destino. Infelizmente, este esquema entra diretamente em conflito com a maneira pela qual a maioria dos firewalls funciona. Uma proposta para desfazer esse recurso se encontra em andamento no IETF.

A Tabela 21.3 mostra o estado das implementações IPSEC para as distribuições usadas como exemplo neste livro. Note que embora o IPSEC rode em cada sistema, nenhuma versão norte-americana o inclui na instalação básica. Para informações sobre disponibilidade e configuração, visite o site www.freeswan.org.

Tabela 21.3 Situação da implementação IPSEC para diversas distribuições Linux.

Sistema	Conseguiu?	Comentários
Red Hat	não muito	O FreeS/WAN IPSEC foi desenvolvido no Red Hat, mas não é distribuído como padrão devido a restrições das leis de exportação dos Estados Unidos
SuSE	sim	FreeS/WAN IPSEC, desde 1999
Debian	mais ou menos	O FreeS/WAN IPSEC faz parte do Gibralta, um firewall Linux; atualmente já estão disponíveis pacotes para Debian. O kernel padrão distribuído com o Debian, entretanto, não tem o suporte a IPSEC; é preciso recompilar o kernel. Também disponíveis estão pacotes para modificação apropriada do kernel e sua compilação.

Note que há uma pegadinha relacionada a túneis IPSEC e tamanho MTU. É importante garantir que, uma vez que um pacote tenha sido criptografado pelo IPSEC, nada o fragmente ao longo do caminho percorrido pelo túnel. Para realizar esse feito, pode ser necessário diminuir o MTU nos dispositivos em frente ao túnel (no mundo real, 1.400 bytes normalmente funciona). Veja a página 184 no capítulo sobre TCP para obter informações adicionais sobre o tamanho de MTU.

Tudo o que eu preciso é uma VPN, não é mesmo?

Infelizmente, há um lado negativo em relação às VPNs. Embora elas realmente forneçam um túnel (em grande parte) seguro entre as duas extremidades, normalmente não se preocupam com a segurança destas extremidades em si. Se, por exemplo, configurarmos uma VPN entre um backbone corporativo e a residência do presidente da empresa, pode ser que estejamos criando inadvertidamente um caminho para a filha de 15 anos deste presidente, fazendo com que esta adolescente tenha acesso direto a tudo o que existe na rede corporativa. Com muita sorte, esta adolescente usará seu recém-adquirido acesso somente para marcar um encontro com um rapaz bonitinho da expedição.

Em suma: você precisa tratar as conexões de túneis VPN como conexões externas e conceder a elas privilégios adicionais somente por absoluta necessidade e após cuidadosa consideração. Talvez seja interessante considerar o acréscimo de um seção especial à política de segurança de sua instalação que abranja quais regras se aplicam a conexões VPN.

21.13 Fontes de Informações sobre Segurança

Metade da batalha para manter um sistema seguro consiste em estar em compasso com os desenvolvimentos relacionados com a segurança como um todo. Se sua instalação for invadida, provavelmente não será através de uma técnica recente. Mais provavelmente, a fissura em sua armadura já foi amplamente discutida em listas de discussão e grupos de notícias relacionados com segurança de sistemas.

www.snow666.blogspot.com
Snow666

CERT: uma marca de serviços registrada da Carnegie Mellon University

Em resposta ao alvoroço sobre o worm Internet de 1988, a DARPA (Defense Advanced Research Projects Agency) formou uma organização chamada CERT (Computer Emergency Response Team) para atuar como uma alfândega para informações sobre segurança em sistemas computacionais. O CERT ainda é o mais conhecido ponto de contacto para informações sobre segurança, embora ele pareça ter crescido um tanto lento e ultimamente, burocrático. O CERT agora também insiste que o nome CERT não significa nada mais do que uma simples "marca de serviços registrada da Carnegie Mellon University".

Embora o "contrato social" do CERT inclua, em certo grau a solução de problemas, na realidade falta ao CERT capacidade para investigar problemas ou para disciplinar transgressores e, portanto, ele nada mais é do que um depósito para correções de segurança e de anúncios de ferramentas para segurança dos fabricantes. Essas correções e os anúncios são chamados "boletins CERT". Novos boletins são postados para www.cert.org, enviados por e-mail à lista de discussão cert-advisory e submetidos ao grupo de notícias comp.security.announce. Para fazer uma assinatura, vá para

http://www.cert.org/contact_cert/certmaillist.html.

SecurityFocus.com e a lista de discussão BugTraq

SecurityFocus.com é um site especializado em notícias e informações relacionadas com segurança. As notícias incluem artigos atuais sobre questões genéricas e sobre problemas específicos; há também uma imensa biblioteca de artigos técnicos úteis, agradavelmente organizados por tópico.

O arquivo de ferramentas de segurança do SecurityFocus inclui software para uma série de sistemas operacionais, juntamente com sinopses e avaliações de usuários. É a fonte mais abrangente e detalhada de ferramentas de que temos conhecimento.

A lista BugTraq é um fórum com moderador para a discussão de vulnerabilidades sobre segurança e suas correções. Para fazer uma assinatura, envie um e-mail para listserv@securityfocus.com com o seguinte corpo de mensagem:

SUBSCRIBE BUGTRAQ *sobrenome*, *nome*

Entretanto, o tráfego nesta lista pode ser um tanto pesado. Um banco de dados dos relatórios de vulnerabilidade do BugTraq também pode ser encontrado no site Web.

Boletim Crypto-Gram

O boletim mensal Crypto-Gram é uma fonte de informações valiosa e algumas vezes divertida, relativa à criptografia e à segurança em computadores. Ele é produzido por Bruce Schneier, autor dos respeitadíssimos livros *Secrets and Lies* e *Applied Cryptography*. Para fazer uma assinatura do Crypto-Gram vá para

http://www.counterpane.com/crypto-gram.html

SANS: System Administration, Networking, and Security Institute

SANS é uma organização profissional que patrocina conferências e programas de treinamento relacionados à segurança, bem como publica uma série de informações relacionadas à segurança. O site Web da SANS, www.sans.org, é um recurso útil que ocupa uma posição intermediária entre o SecurityFocus e o CERT: não tão frenético como o primeiro nem tão enfadonho quanto o último.

O SANS oferece vários boletins semanais e mensais via e-mail para os quais você pode fazer uma assinatura no site Web do SANS. O semanal NewsBites é substancial, ao passo que os boletins mensais contêm muita propaganda. Nem um deles é uma grande fonte de novidades sobre segurança.

Recursos de segurança específicos a cada distribuição

Pelo fato de os problemas de segurança terem potencial para gerar um bocado de má publicidade, os fornecedores normalmente estão ávidos para ajudarem os clientes a manterem seus sistemas seguros. A maioria dos grandes fornecedores possui uma lista de discussão oficial para a qual são postados boletins relacionados com segurança de sistemas e muitos também mantêm um site Web sobre questões relacionadas com segurança. É comum as correções de software relacionadas com segurança serem distribuídas gratuitamente, mesmo por parte de fornecedores que em geral cobram pelo suporte de software.

Há também portais sobre segurança na Web como o www.securityfocus.com, que contém informações específicas por fornecedor e links para os posicionamentos oficiais mais recentes do fornecedor.

Snow666

Uma lista de boletins de segurança do Red Hat pode ser encontrada em www.redhat.com/support/errata. Na época em que este livro era escrito, não havia nenhuma lista de discussão oficial sobre segurança patrocinada pelo Red Hat. Entretanto, há uma série de recursos sobre segurança de sistemas Linux na rede, cuja grande parte se aplica diretamente ao Red Hat.

Você pode encontrar boletins de segurança sobre o SuSE em

http://www.suse.com/us/support/security/index.html

Você pode se associar à lista de discussão oficial do SuSE sobre anúncios de segurança visitando

http://www.suse.de/en/support/mailinglists/index.html

Verifique www.debian.org para ver as notícias mais recentes sobre segurança do Debian ou assine lista de discussão em

http://www.debian.org/MailingLists/subscribe#debian-security-announce

Informações relativas à segurança de produtos Cisco são distribuídas na forma de notas de campo, cuja lista pode ser encontrada em www.cisco.com/warp/public/770. Para assinar lista de discussão sobre segurança da Cisco, envie um e-mail para majordomo@cisco.com com a linha "subscribe cust-security-announce" no corpo da mensagem.

Outras listas de discussão e sites Web

Os contatos listados anteriormente são apenas alguns dos numerosos recursos relacionados com segurança disponíveis na rede. Dado o volume de informações que agora se encontra à disposição e à rapidez com a qual os recursos surgem e desaparecem, achamos mais útil indicar alguns meta-recursos.

Um bom ponto de partida é o site Web X-Force (xforce.iss.net) na Internet Security Systems, que mantém uma série de FAQs úteis. Uma delas é a lista atualizada de listas de discussão relacionadas com segurança. As FAQs sobre correções de segurança e fornecedores fornecem informações de contato úteis para uma série de fabricantes.

www.yahoo.com possui uma enorme lista de links relacionados com segurança de sistemas; não deixe de ver a seção sobre segurança dentro da seção genérica "computers and Internet", pois a seção específica ao Linux é um tanto anêmica. Uma oura excelente fonte de links sobre o assunto de segurança de redes pode ser encontrada em www.about.com.

21.14 Distribuições Linux Fortificadas

Felizmente (?), fomos abençoados com uma série de iniciativas para produzir versões "fortificadas" do Linux que fornecem uma gama mais ampla de recursos de segurança do que é encontrada nas versões comuns, de prateleira. O fortalecimento normalmente assume a forma de controles de acesso especiais e de capacidades de auditoria. Estes recursos são particularmente úteis caso esteja pretendo usar o Linux em algum tipo de produto projetado especificamente para redes. Entretanto, não está claro se fornecem vantagens substanciais aos usuários de versões comuns, de prateleira. Eles ainda requerem boa higiene, uma boa filtragem de pacotes e todas as demais coisas discutidas neste capítulo. Talvez sejam bons para paz de espírito adicional.

A Tabela 21.4 lista alguns dos projetos de fortalecimento mais conhecidos que você poderá visitar para ver o que eles têm a oferecer.

Tabela 21.4 Distribuições Linux fortificadas.

Nome do projeto	Site Web
Bastille Linux	www.bastille-linux.org
Engarde Linux	www.engardelinux.com
Linux Intrusion Detection System	www.lids.org
Security Enhanced Linux da NSA	www.nsa.gov/selinux

21.15 O Que Fazer Quando Sua Instalação Tiver Sido Atacada

O segredo para tratar um ataque é simples: não se desespere. É bem provável que no momento em que você descobrir a invasão, a maior parte dos danos já foi feita. De fato, provavelmente isso vem ocorrendo há semanas ou meses. A chance de você descobrir uma invasão que acabou de acontecer há uma hora atrás é mínima ou nenhuma.

www.snow666.blogspot.com
Snow666

Sob essa perspectiva, a sábia coruja diz para respirar fundo e começar a desenvolver uma estratégia cuidadosamente planejada para lidar com a invasão. Você precisa evitar dar indícios ao invasor anunciando a invasão ou realizando qualquer atividade que poderia parecer anormal para alguém que está observando as operações de sua instalação há várias semanas. Dica: realizar um backup do sistema normalmente é uma boa idéia nesse ponto e (assim esperamos!) parecerá ser uma atividade normal ao invasor.[19]

Isso também serve para relembrá-lo de que alguns estudos revelam que 60% dos incidentes relacionados com segurança envolvem uma pessoa interna. Tome cuidado com quem você discute o incidente até ter certeza de todos os fatos.

Eis um plano de 9 etapas que pode ajudá-lo neste momento de crise:

Passo 1: Não se desespere. Em muitos casos, um problema não é percebido a não ser horas ou dias depois que ele ocorreu. Algumas poucas horas ou dias a mais não afetarão o resultado. A diferença entre uma resposta de quem entrou em pânico e uma resposta racional, por outro lado, afeta o resultado. Muitas situações de recuperação são exacerbadas pela destruição de logs, condições e informações de rastreio importantes durante um pânico inicial.

Passo 2: Decida sobre o nível apropriado da resposta. Ninguém se beneficia de um incidente de segurança superanunciado. Proceda de forma calma. Identifique o pessoal e os recursos que devem participar e deixe os demais ajudarem no *post-mortem* depois de tudo estar terminado.

Passo 3: Guarde como tesouro todas as informações disponíveis de rastreio. Verifique arquivos de contabilidade e de logs. Tente determinar onde a brecha original ocorreu. Realize um backup de todos os seus sistemas. Certifique-se de que protegeu fisicamente contra gravação suas fitas de backup caso as coloque numa unidade de fita para lê-las.

Passo 4: Avalie o seu grau de exposição. Determine que informações críticas (se realmente alguma) "deixou" a empresa e imagine uma estratégia de mitigação apropriada. Determine o nível de risco futuro.

Passo 5: Puxe o plugue. Se necessário e apropriado, desconecte máquinas comprometidas da rede. Feche brechas conhecidas e interrompa a sangria.

Passo 6: Elabore um plano de recuperação. Com um colega criativo, elabore um plano de recuperação numa lousa próxima. Este procedimento é mais eficiente quando realizado longe do teclado. Concentre-se em apagar o incêndio e em minimizar os danos. Evite esbravejar e ou criar excitação. Em seu plano, não se esqueça de visar o abalo psicológico que sua comunidade de usuários possa experimentar.

Passo 7: Anuncie o plano de recuperação. Instrua os usuários e os gerentes sobres os efeitos da invasão, o potencial para problemas futuros e sua estratégia de recuperação preliminar. Seja franco e honesto. Os incidentes relacionados com segurança fazem parte da vida num ambiente em rede. Eles não são reflexo de suas capacidade como administrador de sistemas ou qualquer outra coisa para você ficar embaraçado. Admitir abertamente que você teve um problema representa 90% em ganhar uma batalha, desde que você consiga demonstrar que tem um plano para remediar a situação.

Passo 8: Implemente o plano de recuperação. Você conhece os seus sistemas e redes melhor do que ninguém. Siga seu plano e seu instinto. Fale com um colega numa instituição similar (preferencialmente um que o conheça bastante) para você se manter no caminho correto.

Passo 9: Relate o incidente às autoridades competentes. Se o incidente envolveu terceiros, você deve relatar o problema ao CERT, à polícia, ou órgãos competentes. Forneça o maior número de informações possível. Um formulário-padrão se encontra disponível em www.cert.org para ajudá-lo a refrescar sua memória. Eis algumas das informações úteis que você poderia fornecer.

- Os nomes, tipos de hardware e versões de sistema operacional das máquinas comprometidas
- A lista de correções que foram aplicadas no momento do incidente
- Uma lista das contas sabidamente comprometidas
- Os nomes e endereços IP e quaisquer hosts remotos que foram envolvidos
- Informações de contato, caso você as tenha, para os administradores de instalações remotas
- Entradas de log relevantes ou informações de auditoria

O órgão brasileiro responsável por receber tais denúnias é o NBSO (www.nbso.nic.br). Nesse site há documentos indicando como proceder a reclamação, bem como indicação de quais dados são necessários.

Se você acredita que um problema de software anteriormente não documentado tenha sido envolvido, também deve informar o incidente a seu fornecedor.

19. Se os backups de sistema não forem uma atividade "normal" em sua instalação, você terá problemas muito maiores do que a invasão em si.

Snow666

21.16 Leitura Recomendada

BRYANT, WILLIAM. "Designing an Authentication System: a Dialogue in Four Scenes." web.mit.edu/kerberos/www/dialogue.html.
CERT COORDINATION CENTER. "Intruder Detection Checklist." www.cert.org/tech_tips/intruder_detection_checklist.html.
CERT COORDINATION CENTER. "UNIX Configuration Guidelines." www.cert.org/tech_tips/unix_configuration_guidelines.html.
CHESWICK, WILLIAM R.; STEVEN M. BELLOVIN. *Firewalls and Internet Security, 2. ed.* Reading, MA; Addison-Wesley. 2003.
CURTIN, MATT; MARCUS RANUM. "Internet Firewalls: Frequently Asked Questions." www.interhack.net/pubs/fwfaq.
FARMER, DAN; WIETSE VENEMA. "Improving the Security of Your Site by Breaking Into it." 1993. www.fish.com/security.
FARROW, RIK E RICHARD POWER. *Network Defense article series*. 1998-2001. www.spirit.com/Network.
FRASER, B. (Ed.) *RFC2196: Site Security Handbook*. www.rfc-editor.org.
GARFINKEL, SIMSON; GENE SPAFFORD. *Practical UNIX and Internet Security, 3. ed.* Sebastopol: O'Reilly & Associates. 2003.
KERBY, FRED et al. "SANS Intrusion Detection and Response FAQ." SANS. www.sans.org/resources/idfaq/.
MANN, SCOTT ; ELLEN L. MITCHELL. *Linux System Security: The Administrator's Guide to Open Source Security Tools, 2. ed.* Upper Saddle River, NJ: Prentice Hall PTR. 2002.
MORRIS, ROBERT; KEN THOMPSON. "Password Security: A Case History." Communications of the ACM, 22 (11): 594-597, nov. 1979. Reimpresso no *UNIX SystemManager's Manual*, 4.3 Berkeley Software Distribution. University of California, Berkeley, abr. 1986.
PICHNARCZYK, KARYN; STEVE, WEEBER; RICHARD, FEINGOLD. "UNIX Incident Guide: How to Detect an Intrusion." Computer Incident Advisory Capability, U.S. Department of Energy. 1994. http://ciac.llnl.gov/ciac/documents/CIAC-2305_UNIX_Incident_Guide_How_to_Detect_an_Intrusion.pdf.
RITCHIE, DENNIS M. "On the Security of UNIX." Maio 1975. Reimpresso no *UNIX SystemManager's Manual*, 4.3 Berkeley Software Distribution. University of California, Berkeley, abr. 1986.
SCHNEIER, BRUCE. *Applied Cryptography: Protocols, Algorithms, and Source Code inC*. New York, NY: Wiley, 1995.
THOMPSON, KEN. "Reflections on Trusting Trust." *In ACM Turing Award Lectures: The First Twenty Years 1966-1985*. Reading, MA: ACM Press (Addison-Wesley). 1987.
ZIMMERMANN, PHILIP R. *The Official PGP User's Guide*. Cambridge: MIT Press, 1995.
SONNENREICH, WES; TOM YATES. *Building Linux and OpenBSD Firewalls*. New York, NY: J.W. Wiley. 2000.

Trata-se de um livro pequeno impressionante: é fácil para ser lido, apresenta bons exemplos, tem um bom senso de humor e no geral é simplesmente excelente. Nossa única restrição em relação a este livro é o fato de ser contra o emprego de **sudo** para acesso root, alegando que ele é muito difícil de ser empregado e não valer a pena a dor de cabeça. Nós discordamos redondamente.

21.17 Exercícios

E21.1 Discuta o poder da autenticação SSH com senhas Linux *versus* autenticação SSH com uma frase-senha e um par de chaves. Se uma delas for claramente mais segura do que a outra, você automaticamente deverá requerer o método de autenticação mais seguro?

★ **E21.2** **tripwire** identifica arquivos que foram alterados.

a) O que é necessário para configurar e usar **tripwire** em sua máquina?
b) Contra quais males recentes da Internet **tripwire** seria eficiente?
c) Que males recentes da Internet **tripwire** seria inútil?
d) Dado o acesso físico a um sistema, como **tripwire** poderia ser evitado?
e) À que conclusão você poderia chegar se **tripwire** dissesse que **/bin/login** foi alterado, mas ele parece ter o mesmo tamanho e data de modificação anteriores? O que acontece se o programa **sum** der os mesmos resultados para as versões antiga e nova?

★ **E21.3** O tunelamento SSH normalmente é a única maneira de se tunelar tráfego para uma máquina remota na qual você não tem acesso como administrador. Leia a página de manual **ssh** e forneça uma linha de comando que tunele tráfego da porta host local 113 para a porta 113 de mail.remo-

Snow666

tenetwork.org. O ponto de encaminhamento de seu túnel também deve ser o host mail.remote-network.org.

★ **E21.4** Escolha um incidente de segurança recente e pesquise-o. Localize as melhores fontes de informação sobre o incidente e encontre correções ou meios de evitar que sejam apropriados aos seus sistemas. Liste as fontes e as ações propostas para proteger o seu sistema.

★★ **E21.5** Com permissão de seu grupo de administrador de sistemas local, instale **crack**, o programa que procura logins com senhas frágeis. Modifique o código-fonte de **crack** de forma que ele gere somente os nomes de login com os quais senhas frágeis estejam associadas, não com as senhas em si. Execute **crack** no seu arquivo de senhas local (você precisará ter acesso a **/etc/shadow**) e veja quantas senhas que podem ser quebradas você consegue localizar. Configure sua própria senha para uma palavra de dicionário e dê a **crack** simplesmente sua própria entrada em **/etc/shadow**. Quanto tempo **crack** leva para descobri-la? Tente outros padrões (letras maiúsculas, número após a palavra de dicionário, senha de uma só letra etc.) para ver exatamente quão inteligente **crack** é.

★★ **E21.6** No laboratório de computação, configure duas máquinas: um alvo e um sondador. Instale **nmap** e Nessus na sonda. Ataque o alvo com estas ferramentas. Como você poderia detectar o ataque no alvo? Configure um firewall no alvo usando **iptables** para se defender contra as sondas. Você é capaz de detectar o ataque agora? Como? Que outras defesas podem ser configuradas contra os ataques? (Requer acesso root.)

★★ **E21.7** Uma equipe de segurança localizou recentemente uma grande brecha em diversos servidores **sshd** antigos e recentes. Localize uma boa fonte de informações sobre a brecha e discuta as questões e a melhor maneira de enfrentá-las.

★★ **E21.8** Programas setuid são algumas vezes um mal necessário. Entretanto, aplicar setuid a scripts de shell deve ser evitado. Por quê?

★★ **E21.9** O que as regras na seguinte saída gerada por **iptables** permitem ou não?
Quais seriam algumas adições muito fáceis que ajudariam a otimizar a segurança e privacidade? (Dica: as cadeias `OUTPUT` e `FORWARD` poderiam usar algumas outras regras.)

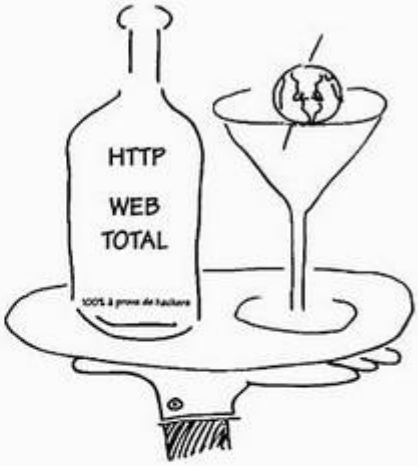
```
Chain INPUT (policy ACCEPT)
Target   prot opt  source    destination
Block    all  --   anywhere  anywhere

Chain FORWARD (policy ACCEPT)
Target   prot opt  source    destination
         all  --   anywhere  anywhere

Chain OUTPUT (policy ACCEPT)
Target   prot opt  source       destination

Chain block (1 references)
Target   prot opt  source          destination
ACCEPT   all  --   anywhere        anywhere     state RELATED,ESTABLISHED
ACCEPT   tcp  --   anywhere        anywhere     state NEW tcp dpt:www
ACCEPT   tcp  --   anywhere        anywhere     state NEW tcp dpt:ssh
ACCEPT   tcp  --   128.138.0.0/16  anywhere     state NEW tcp dpt:kerberos
ACCEPT   icmp --   anywhere        anywhere
DROP     all  --   anywhere        anywhere
```

★★ **E21.10** Inspecione os conjuntos de regras do firewall local. Discuta o que você encontrar em termos de políticas. Há algumas brechas de segurança evidentes? (Este exercício pode requerer a cooperação dos administradores responsáveis para a segurança de seu site local.)

★★★★★ **E21.11** Crie uma ferramenta que determine se qualquer interface de rede em sua instalação está em modo promíscuo. Execute-a regularmente em suas redes para localizar rapidamente tal intrusão. Que nível de carga a ferramenta gera?
Você tem que executá-la em cada máquina ou poderá executá-la remotamente?
Você é capaz de criar um pacote sorrateiro que lhe informaria se uma interface estivesse em modo promíscuo? (Requer acesso root.)

Snow666

22 Hospedagem Web e Servidores Internet



Os últimos anos têm sido uma frenética corrida no mundo da computação. O UNIX foi o imenso reservatório carregado de aminoácidos que deu origem à computação cliente/servidor moderna e à própria Internet. Quando a World Wide Web entrou em cena como o mais recente aplicativo cliente/servidor distribuído no início dos anos 90, o UNIX estava lá como sua plataforma pronta e uma nova era surgia. No final dos anos 90, o Linux se tornou uma popular plataforma para hospedagem devido ao seu custo (ou falta de), velocidade, variedade de plataformas de hardware suportadas e flexibilidade.

Hoje há uma variedade de serviços baseados na Internet que talvez você queira "hospedar" em sua instalação ou em algum dos muitos provedores de terceirização de publicação de sites. Neste capítulo, tratamos dos dois serviços mais comuns: a Web e o FTP.

22.1 Hospedagem Web

No início dos anos 90, o UNIX era (literalmente) a única opção para servir conteúdo para a Web. À medida que a popularidade da Web foi crescendo, um número cada vez maior de participantes — variando de agências de propaganda a zoológicos — desenvolveu interesse em ter suas presenças marcadas na rede. Entretanto, o UNIX e o Linux representavam uma cultura estranha para muitos desses participantes.

Aproveitando a oportunidade, empresas grandes e pequenas entraram na dança com suas próprias soluções de servidor. Em muitos casos, essas soluções envolviam uma reengenharia substancial dos sistemas operacionais que, diferentemente do Linux, não foram criados desde a base tendo em mente ambientes multitarefa preemptivos. Não obstante, surgiu um novo segmento da indústria conhecido como "hospedagem Web" ou "hospedagem Internet" em torno da tarefa de servir conteúdo para a Web. Os servidores de hospedagem Web não somente oferecem páginas Web brutas (HTML) como também oferecem suporte a serviços como FTP, SSL e fluxos de áudio ou vídeo.

Nos dias de hoje temos à disposição uma série de plataformas para hospedagem Web e uma série de servidores Web especializados que foram desenvolvidos para atender às necessidades de mercados específicos. O outrora e futuro carro-chefe da Microsoft, o Windows, foi amplamente promovido como uma plataforma para hospedagem Web. Porém para pessoas que procuram confiabilidade, manutenção, segurança e desempenho extremos, o Linux ainda está à frente da boiada no jogo da hospedagem Web.

A imprensa especializada publicou inúmeros artigos que faziam a seguinte pergunta: "Qual é a melhor plataforma para hospedagem Web?"; comumente, o Windows e o Linux se posicionavam em cantos opostos do ringue. Há razões concretas do porquê o Linux normalmente ser uma melhor opção para ambientes de produção.

Snow666

As maiores vantagens do Linux são sua facilidade de manutenção e desempenho. O Linux foi concebido desde o início como um sistema operacional interativo e multiusuário. Num computador Linux, um administrador pode manter um banco de dados enquanto um outro se preocupa com o desempenho de I/O e um terceiro mantem o servidor Web. No Windows, a pessoa com o controle da console (seja local, como remotamente, usando uma ferramenta como o PC-Anywhere ou o software incluso com o Windows XP) é a única capaz de realizar tarefas administrativas. No que se refere ao desempenho, um bom administrador é capaz de ajustar o Linux para que tenha um desempenho duas ou três vezes mais rápido do que o Windows em hardware idêntico.

22.2 Fundamentos sobre Hospedagem WEB

Hospedar um site Web não é substancialmente diferente de fornecer qualquer outro serviço de rede. A base da World Wide Web é o HTTP (Hyper-Text Transfer Protocol, protocolo de transferência de hipertexto), um protocolo simples baseado em TCP que é usado para formatar, transmitir e conectar documentos contendo uma série de tipos de mídia, entre os quais texto, imagens, som, animação e vídeo. O HTTP se comporta de forma muito similar aos demais protocolos cliente/servidor usados na Internet, por exemplo, o SMTP (para e-mail) e o FTP (para transferência de arquivos).

Um servidor Web nada mais é que um sistema configurado para responder a solicitações HTTP. Para converter seu sistema Linux genérico numa plataforma para hospedagem Web, você precisará instalar um daemon que escute conexões na porta TCP 80 (o padrão HTTP), aceite solicitações de documentos e os transmita ao usuário solicitante.

Navegadores Web como o Netscape, o Opera e o Internet Explorer contactam servidores Web remotos e fazem solicitações em nome dos usuários. Os documentos assim obtidos podem conter ponteiros-hipertexto (links) para outros documentos que podem ou não residir no servidor que o usuário contactou originalmente. Já que o padrão de protocolo HTTP é bem definido, clientes usando qualquer sistema operacional ou arquitetura poderão se conectar ao servidor HTTP. Essa independência de plataforma, juntamente com a capacidade do HTTP de passar um usuário de um servidor a outro de forma transparente, ajudou a incendiar o seu sucesso.

Entretanto, há vida além do HTTP puro e simples. Muitos protocolos otimizados agora foram definidos para fornecer de tudo, de criptografia a fluxo de vídeo. Esses serviços adicionais são normalmente controlados por daemons distintos, mesmo se eles forem fornecidos pelo mesmo servidor físico. Por exemplo, um dos serviços otimizados mais populares é o HTTPS, ou seja, HTTP Seguro. Ele é tratado por um daemon que entende o protocolo SSL (Secure Socket Layer, camada de soquete seguro) e escuta solicitações na porta TCP 443.

Localizadores de recurso uniforme

Um URL (Uniform Resource Locator, localizador de recurso uniforme) é um ponteiro para um objeto ou serviço na Internet. Ele descreve como acessar um objeto por meio de cinco componentes básicos:

- Protocolo ou aplicativo
- Nome do host
- Porta TCP/IP (opcional)
- Diretório (opcional)
- Nome do arquivo (diferencia o uso de letras maiúsculas/minúsculas; normalmente termina em ".htm" ou ".html")

O Diagrama A ilustra um URL típico e seus componentes.

Diagrama A Partes de um URL

Snow666

A Tabela 22.1 ilustra os protocolos comumente usados em URLs.

Tabela 22.1 Protocolos URL.

Protocolo	O que ele faz	Exemplo
http	Acessa um arquivo remoto via HTTP	http://admin.com/index.html
https	Acessa um arquivo remoto via HTTP/SSL	https://admin.com/order.shtml
ftp	Acessa um arquivo remoto via FTP	ftp://ftp.xor.com/adduser.tar.gz
mailto	Envia e-mail para um endereço designado	mailto:sa-book@admin.com
news	Acessa grupos de notícias Usenet	news:alt.cooking
telnet	Loga num computador remoto	telnet://spot.acme.com
ldap	Acessa serviços de diretórios LDAP	ldap://ldap.bigfoot.com:389/cn=Herb
file	Acessa um arquivo local (sem Internet)	file://etc/syslog.conf

Como o HTTP funciona

HTTP é o protocolo que faz com que a World Wide Web realmente funcione e, para a alegria de muitos, ele é um protocolo cliente/servidor sem estado e extremamente básico. No paradigma HTTP, o iniciador de uma conexão sempre é um cliente (normalmente um navegador). O cliente solicita ao servidor o "conteúdo" de um determinado URL. O servidor responde com um fluxo de dados ou com algum tipo de mensagem de erro. No HTTP versões 0.9 e 1.0, a conexão é então fechada; no HTTP 1.1, o cliente pode prosseguir para solicitar um outro objeto.

Pelo fato de o HTTP ser tão simples, você pode facilmente se transformar num navegador Web rudimentar usando **telnet**. Já que a porta-padrão para o serviço HTTP é a porta 80, basta enviar um **telnet** diretamente a esta porta no servidor Web de sua preferência. Assim que estiver conectado, você pode emitir comandos HTTP. O comando mais comum é GET, que solicita o conteúdo de um documento. Normalmente, GET / é o que você deseja, já que ele solicita o documento-raiz (normalmente a homepage) de seja lá qual for o servidor a qual você se conectou. O HTTP diferencia o emprego de letras maiúsculas/minúsculas, de forma que você deve se certificar de que digitou os comandos em letras maiúsculas.

```
% telnet localhost 80
Trying 127.0.0.1…
Connected to localhost.atrust.com.
Escape character is '^]'.
GET /
<o conteúdo de seu arquivo index.html aparece neste ponto>
Connection closed by foreign host.
```

Scripting CGI: gerando conteúdo dinamicamente

Além de servir documentos estáticos, um servidor HTTP é capaz de fornecer ao usuário o conteúdo que foi criado na hora. Por exemplo, se você quisesse fornecer a hora e temperatura atuais aos usuários que estivessem visitando o seu site Web, poderia fazer com que o servidor HTTP executasse um script para obter estas informações. Esse truque agrade-os-nativos é normalmente realizado pelo CGI (Common Gateway Interface, interface de gateway comum).

O CGI não é uma linguagem de programação, mas sim uma especificação que permite a um servidor HTTP trocar informações com outros programas. Mais freqüentemente, os scripts CGI são programas em Perl, Python ou C que foram escritos especificamente para interfacear com um servidor HTTP. Mas realmente, quase todas as linguagens de programação que são capazes de realizar I/O em tempo real são aceitáveis. Imagine só todos aqueles programadores FORTRAN abandonados que agora poderão reaplicar seus conhecimentos na Internet!

Em grande parte, os scripts CGI são preocupação dos desenvolvedores e programadores Web. Infelizmente, scripts CGI colidem com a tarefa do administrador de sistemas numa importante área: segurança. Pelo fato de os scripts CGI terem acesso a arquivos, conexões de rede e outros métodos de transferência de dados de um local a outro, sua execução pode afetar potencialmente a segurança da máquina na qual o servidor HTTP está sendo executado. Em última instância, um script CGI dá a qualquer um a capacidade de executar um programa (o script CGI) em seu servidor. Portanto, os scripts CGI precisam ser tão seguros como qualquer outro programa que possa ser acessado via rede.

Para uma excelente fonte de informações sobre manipulação segura de scripts CGI, veja o arquivo www.w3.org/Security/Faq/www-security-faq.html.

Snow666

Balanceamento de carga

É difícil prever quantos acessos (solicitações de um único objeto como um arquivo de texto ou imagem) ou visualizações de páginas (solicitações de todos os objetos numa única página visível) um único servidor será capaz de manipular. A capacidade exata de um servidor depende do sistema operacional que está sendo executado, a extensão do ajuste do sistema, a arquitetura de hardware do sistema (inclusive subsistemas) e a construção do site (por exemplo, ele é constituído exclusivamente por páginas HTML, ou são chamadas a bancos de dados e cálculos numéricos a serem feitos?). Somente comparações e medidas diretas do site real executando no hardware real são capazes de responder à pergunta "quantos acessos?". Algumas vezes, pessoas que construíram sites similares em hardware similar podem ser capazes de lhe dar uma aproximação grosseira que será útil para fins de planejamento. Em nenhum caso você deve acreditar nos números informados pelos fornecedores de sistemas comerciais.

Isso dito, em vez de contagens de acessos num único servidor, um parâmetro melhor a ser focalizado é a escalabilidade. Certifique-se de que você e a sua equipe de web design tenham um plano que permita dividir a carga de um site com tráfego muito pesado entre vários servidores. A maneira mais fácil de distribuir o tráfego é usar produtos comerciais de hardware de balanceamento de carga como o Big-IP Controller da F5 Networks ou o Alteon ACEswitch (cujas vendas e marketing agora são feitos pela Nortel). Esses produtos distribuem a carga conforme especificado por uma série de parâmetros configuráveis pelo administrador, como tempo de resposta de servidor individual e disponibilidade. O balanceamento de carga acrescenta desempenho e redundância a uma rede — não saia de casa sem ele.

22.3 Instalação de Servidores HTTP

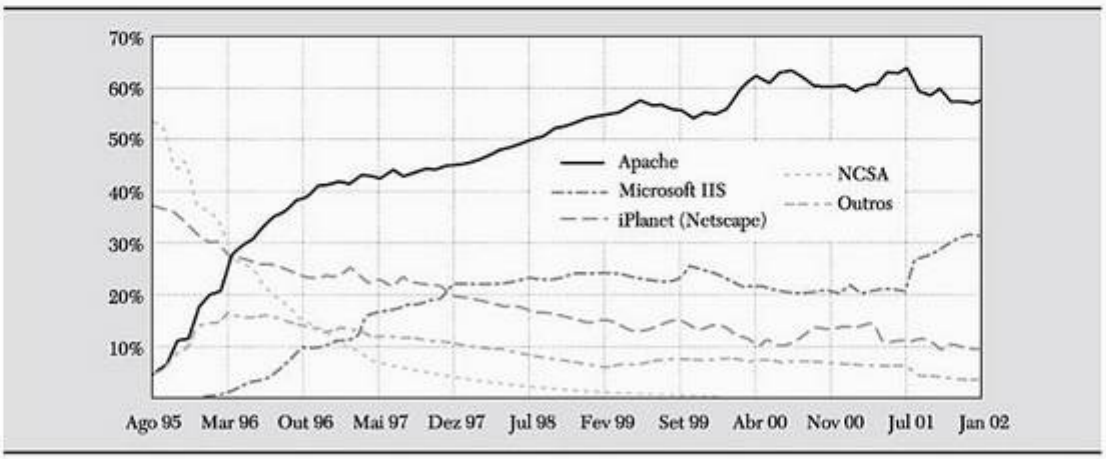
Instalar e manter um servidor Web é fácil! Os serviços Web estão muito abaixo de e-mail e DNS em termos de complexidade e dificuldade de administração.

Escolhendo um servidor

Há vários servidores HTTP excelentes disponíveis para Linux, mas muito provavelmente você optará em começar com o servidor Apache.[1] Esse é o servidor Web padrão dos sistemas Red Hat, SuSE e Debian e é bem conhecido no mercado por sua flexibilidade e desempenho. Por volta de dezembro de 2001, 56.5% dos servidores Web da Internet estavam usando o Apache; em novembro de 2003, o Apache respondeu por 67,41% dos servidores Web da Internet. O Diagrama B mostra as tendências de fatia de mercado ao longo do tempo.

O servidor HTTP Apache é "livre para ser adotado" e o código-fonte completo está disponível no site do Apache Group em www.apache.org. Os menos aventureiros talvez queiram instalar o pacote binário do Apache que faz parte das distribuições Linux. (Mas é provável que eles já estejam instalados em seu sistema.)

Diagrama B Fatia de mercado de servidores Web (todos os domínios)[a]



a. Dados de 12/2001. Cortesia da Netcraft. Visite www.netcraft.com/survey para obter a versão mais recente.

1. O grupo Apache foi formado por várias pessoas que forneciam arquivos de correção para o **httpd** NCSA, um servidor Web popular nos "velhos tempos", por volta de 1993. O resultado final foi um servidor "remendado". O nome Apache vem de um trocadilho: o grupo foi formado por pessoas que forneciam "patch files" para o NCSA; o resultado foi "a patchy" server. Entendeu?

Snow666

Você pode encontrar um útil comparativo de todos os servidores HTTP disponíveis atualmente no site webcompare.internet.com. Eis alguns dos fatores que devem ser considerados em sua decisão:

- Robustez
- Desempenho
- Tempo de espera para atualizações e correções de bugs
- Disponibilidade de código-fonte
- Custo
- Controle de acesso e segurança
- Capacidade de atuar como um proxy
- Disponibilidade de tratar criptografia

Instalando o Apache

Caso opte por fazer o download do código-fonte e compilá-lo por conta própria, você precisará executar o script **configure** incluso com a distribuição. Esse script detecta automaticamente o tipo de sistema usado e configura os makefiles apropriados. Você precisa usar a opção **--prefix** para especificar onde em sua árvore de diretórios o servidor Apache vai residir. Por exemplo:

```
% ./configure --prefix=/etc/httpd/
```

Você pode incluir ou remover de seu servidor alguns dos recursos do Apache invocando as opções **-enable-module=** e **-disable-module=** do script **configure**. Embora o conjunto-padrão de módulos seja razoável, talvez você queira habilitar os módulos mostrados na Tabela 22.2.

Da mesma forma, talvez você queira desabilitar os módulos listados na Tabela 22.3. Para fins de segurança e de desempenho, é uma boa idéia desabilitar módulos que você sabe que não usará.

Tabela 22.2 Módulos Apache úteis que não são habilitados como padrão.

Módulo	Função
auth_dbm	Usa um banco de dados DBM para gerenciar acesso de usuários/grupos (recomendado)[a]
auth_db	Usa um banco de dados DB para gerenciar acesso de usuários/grupos (recomendado)[a]
usertrack	Habilita rastreio rastro-de-clique de navegadores que suportam a tecnologia "cookie"
rewrite	Reescreve URLs usando expressões regulares
expires	Permite que você anexe datas de expiração a documentos
proxy	Usa o Apache como um servidor proxy (maiores informações adiante)

a. Recomendamos que você use um desses módulos, mas não há nenhuma necessidade de usar ambos.

Tabela 22.3 Módulos Apache para os quais sugerimos sua eliminação.

Módulo	Função
asis	Permite que tipos de arquivos designados sejam enviados sem cabeçalhos HTTP
autoindex	Indexa diretórios que não possuem um arquivo HTML padrão (por exemplo, **index.html**)
env	Permite que você configure variáveis de ambiente especiais para scripts CGI
include	Permite inclusões no servidor, um esquema de criação dinâmica de páginas obsoleto
userdir	Permite que usuários tenham seus próprios diretórios HTML

Para uma lista completa dos módulos-padrão, veja o arquivo **src/Configuration** em sua distribuição Apache ou http://www.apache.org/docs/mod/index.html.

Quando **configure** tiver finalizado a execução, execute **make** e, a seguir, **make install** para realmente compilar e instalar os arquivos apropriados.

www.snow666.blogspot.com
Snow666

Configurando o Apache

Assim que o servidor tiver sido instalado, você precisará configurá-lo para o seu aplicativo. Todos os arquivos de configuração são mantidos no diretório **conf** (por exemplo, **/etc/httpd/conf**). Você precisará examinar e personalizar três arquivos de configuração distintos para atender às necessidades de seu site: **httpd.conf**, **srm.conf** e **access.conf**.

httpd.conf especifica como o daemon Apache (**httpd**) interage com o seu sistema. Nesse arquivo, você pode configurar a porta TCP na qual o servidor HTTP escuta consultas (normalmente a porta 80, embora você possa escolher uma outra — e, obviamente, é possível executar vários servidores HTTP em diferentes portas de uma única máquina), a localização de arquivos de log e diversos parâmetros de rede e de desempenho. **httpd.conf** também é o arquivo no qual você pode configurar para que **httpd** responda às conexões de interface virtuais; veja a seção 22.4 para maiores detalhes.

Recursos que o servidor precisa acessar são controlados no arquivo **srm.conf**. Esse arquivo inclui a definição extremamente importante `DocumentRoot`, que define a raiz da árvore de diretórios na qual documentos que podem ser servidos estão localizados. O arquivo também atende a questões como a manipulação de URLS "especiais" como http://www.atrust.com/~steve.

As preocupações em relação à segurança são controladas através do arquivo **access.conf**. Esse arquivo contém diretivas que controlam o acesso por arquivo ou por diretório. As permissões impedem o acesso a arquivos confidenciais via **httpd**, independentemente se for proveniente de dentro ou de fora de seu site.

Você deve especificar ao menos dois controles de acesso: um que cubra o diretório de documentos inteiro e outro que se aplique somente ao diretório **cgi-bin**. Somente o diretório **cgi-bin** especificado deve permitir a execução de scripts. Dessa maneira, usuários individuais não podem criar brechas de segurança — seja de forma acidental ou de outra forma — com seus próprios scripts. Use a opção `ExecCGI` em **srm.conf** para habilitar esta restrição.

Executando o Apache

Você pode iniciar **httpd** manualmente ou por meio de um dos scripts **rc** de sua máquina. A última forma é preferível já que essa configuração garantirá que o servidor Web reinicialize toda vez que a máquina for reinicializada. Para iniciar o servidor manualmente, você digitaria algo como se segue

```
% /usr/sbin/httpd -f /etc/httpd/conf/httpd.conf
```

*Veja o Capítulo 2 para obter maiores informações sobre scripts **rc**.*

Caso queira iniciar **httpd** automaticamente no momento da inicialização, você precisará estabelecer um link em seu diretório **rc** que aponte para o arquivo **/etc/init.d/httpd** (que se encontra instalado como parte do pacote **httpd**). É melhor você iniciar **httpd** posteriormente na seqüência de inicialização, depois de os daemons que controlam funções como roteamento e sincronização de tempo tiverem sido iniciados.

Hospedagem de alto desempenho

Uma das diversas coisas que a comunidade de hospedagem aprendeu ao longo dos últimos anos é que o segredo para alto desempenho é otimizar os servidores para conteúdo estático. O Linux agora dispõe de funções exclusivas nesse aspecto através do servidor Web TUX.

TUX é um servidor Web baseado em kernel que é executado em conjunto com um servidor Web tradicional como o Apache. Sempre que possível, o TUX serve páginas estáticas sem jamais deixar o espaço de kernel, de maneira muito semelhante àquela que **rpc.nfsd** serve arquivos. Essa arquitetura elimina a necessidade de copiar dados entre o kernel e o espaço de usuários e minimiza o número de trocas de contexto. Embora o TUX não seja recomendado para principiantes, ele é uma excelente opção para sites que precisam servir conteúdo com alta velocidade.

Embora o TUX tenha sido desenvolvido pelo Red Hat (e se encontre disponível em www.redhat.com), ele foi lançado sob o GPL e pode ser usado com outras distribuições do Linux. Porém, configurar o TUX pode ser um tanto desafiante. Para detalhes, vá a

http://www.redhat.com/docs/manuals/tux

22.4 INTERFACES VIRTUAIS

Nos primórdios, normalmente uma máquina atuava como o servidor para um único site Web (por exemplo, www.acme.com). À medida que a popularidade da Web cresceu, todo o mundo queria ter um site Web e, da noite para o dia, milhares de empresas se transformaram em provedores de hospedagem Web.

www.snow666.blogspot.com
Snow666

Os provedores rapidamente se deram conta de que eles poderiam ter uma grande economia de escala caso fossem capazes de hospedar mais do que um site num único servidor. Esse truque permitiria que www.acme.com, www.ajax,com, www.toadranch.com e muitos outros sites fossem servidos transparentemente pelo mesmo hardware. Em resposta a essa necessidade comercial nasceram as interfaces virtuais.

As interfaces virtuais permitem que um daemon identifique conexões baseadas não somente no número da porta de destino (por exemplo, porta 80 para HTTP), como também no endereço IP de destino da conexão. Hoje em dia, as interfaces virtuais são usadas amplamente e provaram ser úteis para outros aplicativos além da hospedagem Web.

A idéia é simples: uma única máquina Linux responde na rede a mais endereços IP do que o número de interfaces de rede físicas que ela tem. Cada uma das interfaces de rede "virtuais" resultantes pode ser associada a um nome de domínio correspondente que os usuários na Internet poderão querer se conectar. Esse recurso permite que uma única máquina Linux atenda, literalmente, centenas de sites Web. (Por comparação, um sistema operacional competidor baseado em Intel suporta interfaces virtuais, mas somente pode ser usado na prática para hospedar uma dezena de sites. Obviamente, jamais poderíamos citar nomes.)

O protocolo HTTP 1.1 define uma espécie de funcionalidade similar às interfaces virtuais (oficialmente chamada de "interfaces virtuais não-IP") que elimina a necessidade de atribuir endereços IP exclusivos a servidores Web ou de configurar uma interface especial em nível de sistema operacional. Essa abordagem conserva os endereços IP e é útil para alguns sites, especialmente sites nos quais um único servidor é abrigo para centenas ou milhares de homepages (como, por exemplo, as universidades). Entretanto, o esquema não é muito prático para sites comerciais, ele reduz a escalabilidade (você tem alterar o endereço IP do site para transferi-lo para um outro servidor) e também pode ter um impacto negativo sobre a segurança (caso você filtre o acesso a um site em seu firewall de acordo com os endereços IP). Parece que interfaces virtuais verdadeiras ainda estarão por aí durante um tempo.

Configurando interfaces virtuais

Configurar uma interface virtual envolve duas etapas. Em primeiro lugar, você tem de criar a interface virtual em nível TCP/IP. Em segundo, você tem de informar o servidor Apache sobre as interfaces virtuais que instalou. Tratamos dessa segunda etapa na seção 22.4.

As interfaces virtuais no Linux recebem nomes segundo a seguinte convenção: *interface:instância*. Por exemplo, se sua interface Ethernet for eth0, então as interfaces virtuais associadas a ela seriam eth0:0, eth0:1 e assim por diante. Todas as interfaces são configuradas com o comando **ifconfig**. Por exemplo, o comando

```
# ifconfig eth0:0 128.138.243.150 netmask 255.255.255.192 up
```

configura a interface eth0:0 e atribui a ela um endereço na rede 128.138.243.128/26. Para tornar permanentes as atribuições de endereços virtuais, modifique os arquivos de inicialização de sistema.

Para o Red Hat, você precisará criar um arquivo distinto para cada interface virtual no diretório **/etc/sysconfig/network-scripts**. Por exemplo, o arquivo **ifcfg-eth0:0** correspondente ao comando **ifconfig** mostrado anteriormente conteria

```
DEVICE=eth0:0
IPADDR=128.138.243.150
NETMASK=255.255.255.192
NETWORK=128.138.243.128
BROADCAST=128.138.243.191
ONBOOT=yes
```

Em sistemas SuSE você poderia criar interfaces virtuais com YaST ou então editar manualmente o script **/etc/rc.config** para criá-las. Por exemplo, seu arquivo **rc.config** poderia conter as seguintes entradas para definir tanto eth0 quanto eth0:0

```
# IP Adresses

IPADDR_0="128.138.243.149"
IPADDR_1="128.138.243.150"

# network device names (e.g. "eth0")

NETDEV_0="eth0"
NETDEV_1="eth0:0"
```

Snow666


```
# ifconfig parameters

IFCONFIG_0="128.138.243.149 broadcast 128.138.243.191 netmask
   255.255.255.192"
IFCONFIG_1="128.138.243.150 broadcast 128.138.243.191 netmask
   255.255.255.192"
```

A abordagem do Debian é similar à do Red Hat, mas as definições de interface têm de aparecer no arquivo **/etc/network/interfaces**. As entradas correspondentes à interface eth0:0 em nosso exemplo ficariam

```
iface eth0:0 inet static
   address 128.138.243.150
   netmask 255.255.255.192
   broadcast 128.138.243.191
```

Informando o Apache sobre a existência de uma interface virtual

Além de criar as interfaces virtuais com **ifconfig**, você precisará informar ao Apache quais documentos servir quando um cliente tentar estabelecer uma conexão em cada uma das interfaces. Isso é feito através de uma cláusula `VirtualHost` no arquivo **httpd.conf**, uma cláusula `VirtualHost` para cada interface virtual configurada. Eis um exemplo:

```
<VirtualHost 128.138.243.150>
ServerAdmin webmaster@www.company.com
DocumentRoot /var/www/htdocs/company
ServerName www.company.com
ErrorLog logs/www.company.com-error_log
TransferLog logs/www.company.com-access_log
</VirtualHost>
```

Um cliente que se conecta ao host virtual 128.138.243.150 será servido com documentos do diretório **/var/www/htdocs/company**, que deve ser exclusivo a este site.

22.5 Caching e Servidores Proxy

A Internet e as informações nela contida estão crescendo rapidamente. Portanto, a largura de banda e os recursos computacionais necessários para suportá-la também estão crescendo rapidamente. Como esse estado de coisas pode continuar?

A única forma de lidar com tal crescimento é usar replicação. Independentemente de estar num nível nacional, regional ou de site, o conteúdo Internet precisa estar mais prontamente disponível por meio de uma fonte mais próxima à medida que a Internet vai crescendo. Não faz o menor sentido transmitir a mesma página Web popular da Austrália através de um link muito caro para a América do Norte milhões de vezes ao dia. Deveria haver uma maneira de armazenar esta informação uma vez que ela tenha sido enviada através do link uma vez. Felizmente, há uma maneira.

Uma resposta seria o pacote gratuito Squid Internet Object Cache.[2] Esse pacote é, ao mesmo tempo, um servidor proxy e de caching que roda sob o Linux e suporta diversos protocolos, entre os quais HTTP, FTP e SSL.

Eis como ele funciona. Navegadores Web instalados em clientes (como o Netscape e o Internet Explorer) contactam o servidor Squid para solicitar um objeto da Internet. O servidor Squid faz então uma solicitação em nome do cliente (ou fornece o objeto de seu cache, conforme discutido no parágrafo seguinte) e retorna o resultado ao cliente. Os servidores proxy desse tipo são normalmente usados para aumentar a segurança e para filtrar conteúdo.

Num sistema baseado em proxy, somente uma máquina precisa ter acesso direto à Internet através do firewall da organização. Em organizações como escolas K-12, um servidor proxy também pode filtrar conteúdo para que material inapropriado não caia em mãos erradas. Diversos servidores proxy comerciais e gratuitos (alguns baseados no Squid, outros não) estão disponíveis hoje.

O serviço proxy é ótimo, mas são os recursos de caching do Squid que realmente merecem nossa excitação. O squid não somente coloca em cache informações de solicitações de usuários locais, como também permite que seja criada uma hierarquia de servidores Squid.[3] Grupos de servidores Squid usam o ICP (Internet Cache Protocol, protocolo de cache Internet) para comunicar informações sobre o conteúdo de seus caches.

2. Por que "Squid" (lula)? De acordo com a FAQ, "todos os nomes bons já haviam sido usados".
3. Infelizmente, alguns sites marcam todas as suas páginas como sendo não "cacheável", impedindo o Squid a operar sua mágica. De uma maneira similar, o Squid não é capaz de colocar em cache páginas geradas dinamicamente.

Snow666

Esse recurso permite que os administradores construam um sistema no qual usuários locais contactem um servidor de cache no site para obter conteúdo da Internet. Se um outro usuário naquele site tiver solicitado o mesmo conteúdo, poderá ser retornada uma cópia em velocidade de LAN (normalmente, 10 ou 100 Mb/s). Se o servidor Squid local não a tiver, talvez ele contacte o servidor de caching regional. Como no caso local, se alguém da região tiver solicitado o mesmo objeto, ele será servido imediatamente. Caso contrário, talvez o servidor de caching do país ou continente poderá ser contactado e assim por diante. Os usuários percebem uma melhora de desempenho; portanto, eles ficam contentes.

Para muitos, o Squid oferece benefícios econômicos. Pelo fato de os usuários tenderem a compartilhar suas descobertas na Web, duplicação significativa de solicitações Web externas podem ocorrer num site de porte razoável. Um estudo demonstra que manter um servidor de caching pode reduzir as necessidades de largura de banda em cerca de 40%. Essa eficiência extra pode ser uma grande vantagem em sites que pagam por uso por minuto ou por megabyte. Infelizmente, o uso efetivo do Squid pode exigir que você obrigue todos os seus usuários a configurar seus desktops para usar o servidor Squid como servidor proxy — nem sempre uma tarefa fácil.

Configurando o Squid

O Squid é fácil de ser instalado e configurado. Já que o Squid precisa de espaço para armazenar o seu cache, você deve executá-lo numa máquina dedicada que possua muita memória e espaço de disco livre. Uma configuração razoável poderia ser uma máquina com 256MB de RAM e 20GB de disco.

Você pode obter o pacote Squid em formato RPM ou **apt-get** de sua distribuição ou então fazer o download de uma cópia novinha em folha do Squid em www.squid-cache.org. Se optar pelo caminho faça-você-mesmo, precisará executar o script **configure** no alto da árvore de origem após descompactar a distribuição. Esse script assume que você quer instalar o pacote em **/usr/local/squid**. Caso prefira alguma outra localização, use a opção **- -prefix=***dir* do **configure**. Depois de **configure** ter sido completado, execute **make all** e, a seguir, **make install**.

Assim que tiver instalado Squid, você deve localizar o arquivo de configuração **squid.conf**. Veja o arquivo **QUICKSTART** no diretório de distribuição para uma lista das alterações que devem ser feitas no arquivo **squid.conf** original.

Você também tem de executar **squid -z** manualmente para criar e zerar a estrutura de diretórios na qual serão armazenadas as páginas Web em cache. Finalmente, você pode iniciar o servidor manualmente através do script **RunCache**; eventualmente, você preferirá chamar esse script por meio dos arquivos **rc** de seu sistema para que eles iniciem o servidor Squid quando a máquina for inicializada.

Para testar o Squid, configure o seu navegador Web para usar o servidor Squid como proxy. Essa opção é normalmente encontrada no painel de preferências do navegador.

22.6 Configuração de Servidor FTP Anônimo

FTP é um dos mais antigos e básicos serviços da Internet, embora ele ainda continue a ser usado largamente hoje em dia. Embora o FTP possua uma variedade de empregos internos num site, o emprego mais comum na Internet é o "FTP anônimo", que permite aos usuários que não tenham contas no seu site fazerem download de arquivos que você tornou disponíveis.

O FTP é útil para distribuir correções de bug, software, documentos preliminares e coisas do gênero. Sua principal vantagem em relação ao HTTP (para esta finalidade) é o fato dele permitir que os usuários inspecionem a árvore de materiais disponíveis por conta própria e vejam os tamanhos e as datas de modificação dos arquivos. Você não precisa criar nenhum código em HTML para apontar para arquivos novos — basta jogá-los no lugar certo e pronto.

Para habilitar FTP anônimo, crie uma conta para o falso usuário "ftp", configure seu diretório de usuário e configure o daemon de servidor FTP, **ftpd**. Devido à natureza pública do FTP anônimo, é importante configurá-lo corretamente para que arquivos confidenciais não se tornem acidentalmente disponíveis para todo o mundo.

*Consulte a seção 28.3 para obter maiores informações sobre **inetd**.*

ftpd é controlado por **inetd** e, conseqüentemente, deve ter uma entrada nos arquivos **/etc/inetd.conf** e **/etc/services**. (Se o seu sistema usa **xinetd** em vez de **inetd**, você precisará criar um arquivo em **/etc/xinetd.d** para **ftpd** em vez de criar uma entrada **/etc/inetd.conf**.) Quando um usuário FTP faz o login anônimo, **ftpd** executa uma chamada de sistema **chroot** para tornar invisíveis e inacessíveis os arquivos externos ao diretório **~ftp**. A segurança adicional fornecida por essa precaução é importante, pois **ftpd** tem de executar setuid em root para manipular portas de soquetes privilegiadas.

Snow666

Para permitir **ftp** anônimo de dentro de seu site, tome as seguintes providências:

- Adicione o usuário "ftp" ao seu arquivo de senhas regular.
- Crie subdiretórios **bin**, **etc**, **lib** e **pub** abaixo de **~ftp**.
- Copie o programa **ls** para o diretório **~ftp/bin**.
- Copie ou faça um hard-link das bibliotecas compartilhadas das quais **ls** precise para **~ftp/lib**.[4]
- Copie **/etc/passwd** e **/etc/group** para **~ftp/etc**.
- Edite os arquivos **passwd** e **group** conforme descrito a seguir.
- Substitua todas as senhas em **~ftp/etc/passwd** por asteriscos.
- Configure as permissões apropriadas nos arquivos e diretórios sob **~ftp**.

Ninguém precisa fazer o login na conta ftp; portanto, use um asterisco como senha para ftp. Também é uma boa idéia especificar **/bin/false** como shell de login para ftp.

Já que uma sessão **ftp** anônima executa com **chroot** aplicado em relação a **~ftp**, os subdiretórios **bin** e **etc** devem fornecer uma cópia de todos os comandos e informações de configuração exigidos por **ftpd**. Após o **chroot**, **~ftp/bin** e **~ftp/etc** irão se mascarar como **/bin** e **/etc**.

Na maioria dos casos, **ftpd** usa somente o comando **ls** e cópias-esqueleto de **/etc/passwd** e **/etc/group** por meio de **~ftp/etc**.

Consulte a seção 21.4 para obter maiores informações sobre segurança de senhas.

O arquivo **passwd** sob **~ftp** deve conter somente os usuários root, daemon e ftp. Você deve substituir as senhas por asteriscos já que essa cópia do arquivo **passwd** estará disponível às pessoas que usam seu servidor **ftp**. Mesmo que as senhas sejam criptografadas, ainda há um risco de que outras pessoas as decodifiquem.

Para maior segurança, torne **~ftp/bin/ls** somente executável configurando seu modo para 111. Esse ajuste impede que clientes copiem o binário e o estudem, em busca de pontos fracos.

Coloque os arquivos que deseja tornar disponíveis em **~ftp/pub**.

As permissões nos diversos arquivos e diretórios são extremamente importantes. Recomendamos que as permissões sejam configuradas conforme mostrado na Tabela 22.4.

Tabela 22.4 Permissões recomendadas sob ~ftp.

Arquivo/Diretório	Proprietário	Modo	Arquivo/Diretório	Proprietário	Modo
~ftp	root	555	**~ftp/etc/passwd**	root	444
~ftp/bin	root	555	**~ftp/etc/group**	root	444
~ftp/bin/ls	root	111	**~ftp/pub**	root	755
~ftp/etc	root	555	**~ftp/lib**	root	755

Note que nas distribuições Red Hat e SuSE, o FTP anônimo é ativo por padrão. Embora a configuração-padrão facilite a configuração de FTP anônimo, ela realmente coloca um risco para o administrador de sistemas insuspeito que não está ciente de sua presença. Remova o usuário ftp do arquivo **passwd** caso não queira fornecer serviço FTP anônimo.

Um dos maiores riscos de segurança do FTP anônimo é resultante da permissão a usuários de depositarem arquivos em diretórios FTP. Diretórios que podem ser gravados por qualquer um, independentemente de quão obscuro, rapidamente se tornarão "ninhos" onde os hackers e as crianças pretendendo comercializar suas bugigangas armazenarão seus arquivos, consumindo toda a sua largura de banda e colocando-o diretamente no meio de uma cadeia de atividades que provavelmente é indesejável, se não for ilegal. Não faça parte do problema; jamais permita diretórios FTP anônimos graváveis em seu sistema.

4. Verifique a documentação da sua distribuição para descobrir quais arquivos são necessários; as instalações-padrão do Red Hat, SuSE e do Debian já realizam esta etapa para você. Note que o hard linking funciona somente se os arquivos residirem na mesma partição de disco.

Snow666

22.7 EXERCÍCIOS

★ **E22.1** Configure uma interface virtual em sua estação de trabalho. Execute **ifconfig** antes e depois para ver o que mudou. Você é capaz de efetuar um ping na interface virtual por meio de uma outra máquina na mesma sub-rede? E de uma rede diferente? Por que sim ou por que não? (Requer acesso root.)

★ **E22.2** Com um analisador de pacotes (**tcpdump**) capture uma conversação HTTP bidirecional que faça o upload de informações (por exemplo, preenchendo um formulário ou campo de pesquisa). Comente a sessão para mostrar como o seu navegador transmitiu informações para o servidor Web. (Requer acesso root.)

★ **E22.3** Use um analisador de pacotes para capturar o tráfego quando você abre uma página Web com muito tráfego, como a homepage da amazon.com ou cnn.com. Quantas conexões TCP distintas são abertas? Quem as inicia? Essa é a maneira mais eficiente de usar o protocolo TCP? (Requer acesso root.)

★ **E22.4** Localize arquivos de log de um servidor Web acessível via Internet, talvez o servidor principal de seu site. Examine os arquivos de log. O que você pode dizer sobre os padrões de acesso após um período de algumas horas? Que erros surgiram durante esse período? Quais são as preocupações em relação à segurança indicadas pelo conteúdo dos arquivos de log? (Pode ser que seja necessário acesso root.)

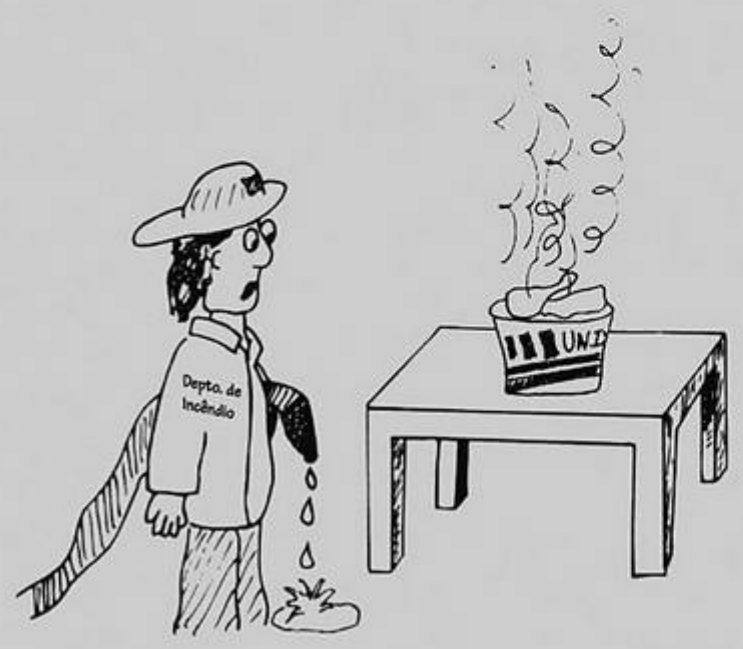
★★ **E22.5** Instale o Apache em seu sistema e crie algumas páginas de conteúdo. De outras máquinas, verifique se o seu servidor Web está operando. Localize os arquivos de log do Apache que permitem que você veja quais navegadores estão acessando o seu servidor. Configure o Apache para servir parte de suas páginas de conteúdo para a interface virtual criada em E22.1. (Requer acesso root.)

★★ **E22.6** Use **tcpdump** para capturar tráfego FTP tanto para sessões FTP ativas quanto passivas. Como a necessidade de suportar um servidor FTP anônimo afeta a política de firewall de um site? O que as regras de firewall precisariam permitir? (Requer accesso root.)

www.snow666.blogspot.com
Snow666

SEÇÃO 3

UM BOCADO DE COISAS



www.sn0w666.blogspot.co
Sn❄w666

Página em branco

Snow666

23 Instalação e Localização de Software

O Linux não aparece simplesmente num computador como num passe de mágica; alguém precisa colocá-lo lá. Embora um número de fabricantes cada vez maior esteja oferecendo sistemas Linux pré-configurados nos dias de hoje, é inevitável que em algum momento você precisará instalar o Linux por conta própria. Da mesma forma, quase sempre você precisará fazer algumas alterações para tratar necessidades particulares de seu ambiente local. Os administradores normalmente têm de realizar todas as tarefas abaixo:

- Instalar o Linux
- Automatizar instalações de massa
- Localizar sistemas
- Manter atualizados os sistemas
- Gerenciar pacotes de software

Este capítulo explora algumas técnicas e aplicativos que ajudam a reduzir a dor de cabeça da instalação de software e faz com que estas tarefas cresçam em escala efetivamente.

23.1 Instalação Básica do Linux

Em grande parte, a instalação básica de um computador Linux é bem objetiva. Fazemos a inicialização por meio de um CD-ROM ou de um disquete, respondemos algumas perguntas, e o instalador da distribuição copiar os pacotes que selecionamos para o nosso disco rígido. Quase todas as distribuições possuem excelentes guias de instalação. Um conjunto útil de guias de instalação independentes pode ser encontrado em http://docs.linux.com.

A versão pronta do Red Hat inclui o manual de instalação. Ele também pode ser encontrado em:

http://www.redhat.com/docs/manuals/linux.

A versão pronta do SuSE também inclui manuais de instalação. O SuSE não parece ser muito bom no que diz respeito à documentação on-line, talvez numa tentativa de motivar a venda da versão no varejo. Uma pena.

O manual de instalação do Debian se encontra em

http://www.debian.org/releases/stable/installmanual.

Snow666

23.2 Instalação Automática

Se alguma vez você já instalou mais do que um computador ao mesmo tempo, achará que rapidamente atinge os limites de instalação interativa. Para início de conversa, ela não é muito reproduzível. Por exemplo, você se lembra exatamente quais pacotes devem ser instalados? Esse tipo de problema pode ser evitado com uma lista de verificação de localização (veja a seção 23.3, mas até mesmo uma lista de verificação não solucionará todos os problemas.

A instalação interativa consome muito tempo e esforço. Mesmo que você possa contratar um número suficiente de pessoas para instalar 1.000 máquinas manualmente, há aplicações melhores para este dinheiro.

Os instaladores para todas as distribuições usadas como exemplo neste livro podem ser escritos em script, de modo que uma outra opção seria criar um disquete contendo a configuração desejada. Essa solução resolve o problema da reprodução, porém ainda não é suficientemente escalável. Se você estiver instalando centenas de máquinas, criar e usar centenas de disquetes de configuração é o cúmulo do tédio.

O Kickstart do Red Hat e o YaST do SuSE (ambos os quais são métodos avançados de criar scripts por meio dos instaladores-padrão das distribuições) podem ser utilizados de maneira que suportem instalação automática pela rede. Há também vários projetos de fonte aberto sedentos para produzirem instaladores de rede de terceiros, por exemplo, o SystemImager.

Os usuários do Debian podem se sentir um pouco ofendidos caso não vejamos o instalador do Debian. Embora ele nominalmente suporte NFS como uma fonte de pacotes, o instalador do Debian infelizmente não possui diversos recursos necessários para suporte apropriados de instalações de rede. Uma total reformulação do instalador se encontra em andamento. Um projeto chamado FAI (Fully Automatic Installation, instalação totalmente automática) pretende fornecer automação para instalações Debian. Veja o endereço

http://www.informatik.uni-koeln.de/fai

para obter maiores informações. De forma alternativa, considere o emprego de um instalador de terceiros como o SystemImager (veja a seção 23.2). O FAI parece ter atingido a maturidade, e os relatórios de utilização indicam que ele tem sido utilizado eficientemente em grandes "clusters" com até 120 nós.

Inicialização de PCs via rede (netbooting)

A inicialização de PCs via rede permite que inicializemos um computador totalmente através da rede, em vez de por meio de um disco rígido, disquete ou CD-ROM. É muito conveniente para a instalação, pois isso significa que você poderia sentar-se à frente de sua mesa e fazer a inicialização via rede de uma máquina em algum outro ponto qualquer sem ter de se deslocar até esta máquina e inserir um CD-ROM ou disquete. Infelizmente, a técnica de rede suportada pela maioria das distribuições é fazer a inicialização por meio de um disquete ou de um CD-ROM e depois obter tudo o mais da rede.

O projeto Etherboot (etherboot.sourceforge.net) cria PROMs de inicialização de fonte aberto para NICs (Network Interface Card, placa de interface de rede) comumente usadas que suportam inicialização somente de rede. Entretanto, já que a maioria das pessoas não tem gravadores de EEPROM em seus escritórios, a utilidade real desta solução não é clara. O suporte está sendo deixado em preferência a um novo padrão aberto para inicialização via rede em sistemas baseados em processadores Intel: o PXE.

PXE: o padrão para inicialização de PCs via rede

PXE (Preboot Execution Environment) é um padrão um tanto engenhoso que abarca e estende o DHCP para oferecer um recurso de inicialização via rede para o PC. O PXE está sendo promovido ativamente pela Intel e parece estar gradualmente dominando o mercado. A maioria dos PCs suporta este padrão. Embora o PXE não seja perfeito, um padrão rudimentar é melhor do que nenhum.

O protocolo PXE é objetivo (e similar aos procedimentos de inicialização via rede usados em outras arquiteturas). Um host faz a difusão ampla de uma solicitação de "descoberta" DHCP com a opção PXE ativa, e um servidor ou proxy DHCP responde com um pacote DHCP que inclui opções PXE (o nome de um servidor de inicialização e um arquivo de inicialização). O cliente faz o download de seu arquivo de inicialização via TFTP (ou, opcionalmente, TFTP multidifusão) e a seguir o executa.

Embora a especificação PXE não seja totalmente compatível com DHCP genérico, na prática muitos servidores DHCP modernos são capazes de fornecer serviço de inicialização PXE. Até mesmos servidores que não suportam o PXE diretamente são capazes de interagir com ele por meio de um mecanismo proxy.

Snow666

Vários sistemas de inicialização via rede baseados em PXE estão em desenvolvimento, inclusive o PXELinux (uma extensão do Syslinux, veja syslinux.hackerdojo.com/pxe.php) e o PXEGRUB (uma extensão do carregador de inicialização GRUB, veja www.gnu.org). A Intel também fornece um pacote PXE para Linux:

http://www.intel.com/ial/wfm/index.htm

Configurando PXE para Linux

O servidor DHCP ISC (Internet Software Consortium), que vem com as distribuições usadas como exemplo neste livro, suporta o PXE. Se você estiver usando um servidor DHCP diferente, instale o proxy do pacote PXE da Intel.

Infelizmente, as distribuições atuais do Linux não integram PXE além do ponto de fornecer suporte DHCP no servidor. Elas oferecem até menos suporte para emprego do PXE como um cliente; os instaladores normalmente não oferecem suporte a ele da forma que vêm de fábrica. Dos instaladores que descrevemos neste capítulo, somente o SystemImager reconhece PXE.

Não obstante, fazer a inicialização via rede é a maneira correta de ser adotada em grandes instalações e você não tem de ficar esperando os distribuidores fazerem com que eles atuem juntos. Documentos que descrevem o PXE estão à disposição em uma série de fontes. Um bom local para iniciar é o *Remote Boot HOWTO* em www.linux-doc.org.

Embora o pessoal do Red Hat ainda não suporte oficialmente o PXE, eles efetivamente disponibilizam alguma coisa para o PXE e distribuem o software Intel PXE como parte de suas distribuições. Kickstart pode ser transformado para operar via PXE; veja

http://www.stanford.edu/~alfw/PXE-Kickstart

O SuSE não vem com nenhum pacote PXE, mas você pode pegar emprestado aqueles do Red Hat. O YaST pode ser adaptado facilmente para operar com PXE, embora esta configuração não seja oficialmente suportada.

O novo Debian Installer, atualmente em beta 1, já tem suporte para inicializar utilizando PXE. A página do projeto Debian Installer http://www.debian.org/devel/debian-installer/ leva a mais documentação sobre como instalar o Debian por meio do PXE.

Fazendo a inicialização via rede de máquinas diferentes de PCs

O PXE é um produto da Intel e é limitado ao hardware IA-32 e IA-64. Outras arquiteturas possuem seus próprios métodos de fazer a inicialização através da rede (quase sempre mais elegante do que o PXE). A discussão das diversas diferenças entre as arquiteturas realmente está fora do escopo deste livro, porém uma série de recursos na Web poderão ajudá-lo.

Computadores Alpha geralmente usam SRM para inicializar Linux, e o *SRM HOWTO* descreve como usar este recurso. Máquinas SPARC e a maioria dos computadores PowerPC usam o Open Firmware, que é fácil de ser inicializado via rede (digite **boot net**). A FAQ UltraLinux (www.ultralinux.org) possui um excelente guia para inicialização do Linux via rede em processadores SPARC. Para Macs, RS/6000s e outras máquinas baseadas no PowerPC, o procedimento para inicialização via rede é específico tanto ao hardware quanto ao carregador de inicialização que está sendo usado, de modo que você deve verificar a documentação de seu carregador de inicialização.

Kickstart: Instalador automático do Red Hat

O Kickstart é a ferramenta do Red Hat para realizar instalações automatizadas. O Kickstart é, na verdade, uma interface de scripting para o instalador do Red Hat, o Anaconda, e é dependente tanto da distribuição Red Hat quanto de pacotes RPM. Diferentemente do SystemImager e outros copiadores de imagens brutos, o Kickstart é flexível e bem inteligente em relação a autodetectar o hardware de um sistema.

Consulte a seção 23.5 para obter maiores informações sobre pacotes.

Caso precise de muita localização, você pode fazer quaisquer alterações que forem necessárias como parte de um script de pós-instalação. Isso funciona, porém os scripts de pós-instalação tendem a se tornar grandes e difíceis de serem mantidos. Uma outra opção é criar um pacote RPM contendo suas personalizações locais. Recomendamos a rota RPM; ela facilita a mudança de versão, facilita futuras atualizações e lhe fornece suporte e todos os demais atrativos que vêm com um sistema na forma de pacote.

www.snow666.blogspot.com
Snow666

Caso queira efetuar grandes mudanças na maneira pela qual o Red Hat vem de fábrica, provavelmente achará o Kickstart abaixo do ideal para as suas necessidades. Nessa situação, provavelmente será melhor usar uma ferramenta de instalação como o SystemImager.

Definindo um arquivo de configuração do Kickstart

O comportamento do Kickstart é controlado por um único arquivo de configuração, geralmente chamado **ks.cfg**. O formato desse arquivo é bem objetivo. Se você tem uma tendência para o visual, o Red Hat inclui uma prática interface gráfica com o usuário chamada **ksconfig** que lhe permite apontar e clicar para o paraíso **ks.cfg**.

O arquivo **ks.cfg** também é bem fácil de ser gerado via programa. Suponha, por exemplo, que você queira instalar um conjunto de pacotes diferente em servidores e clientes e que também precise tratar de instalações ligeiramente diferentes personalizadas para várias filiais. Você poderia criar um pequeno script em Perl que usasse um conjunto mestre de parâmetros para gerar um arquivo de configuração para os servidores e clientes em cada filial. Alterar o complemento de pacotes se tornaria simplesmente uma questão de alterar esse script Perl em vez de alterar cada arquivo de configuração. Pode até haver alguns casos nos quais você precise gerar um arquivo de configuração individualizado para cada host. Nessa situação, certamente você iria querer que os arquivos fossem gerados automaticamente.

Um excelente manual para o Kickstart (incluindo uma lista de todas as opções disponíveis) se encontra no *The Official Red Hat Linux Customization Guide* para o Red Hat: http://www.redhat.com/docs/manuals/linux/

Um pequeno alerta em relação ao arquivo de configuração: se você cometer um erro, o diagnóstico consiste em uma mensagem de rastreio do Python praticamente ininteligível, que pode ou não conter uma mensagem oculta que lhe aponta diretamente o erro. É mais fácil usar **ksconfig**, que deveria gerar um arquivo de configuração que ao menos é sintaticamente válido (embora quando tentamos fazer isto, os comandos de partição que ele nos gerou não funcionaram ...).

Um arquivo de configuração do Kickstart é formado de três partes ordenadas. A primeira delas é a seção de comandos, que especifica opções como a linguagem, o teclado e o fuso horário. Essa seção também especifica a origem da distribuição com a opção `url` (no exemplo a seguir, há um host chamado installserver).

Eis um exemplo de uma seção de comandos completa:

```
text
lang en_US                  # lang é usada durante a instalação...
langsupport en_US           # ...e langsupport em tempo de execução.
keyboard us                 # Usa um teclado Americano.
timezone --utc America/EST  # --utc significa que o relógio do hardware está configurado como UTC (GMT)
mouse
rootpw qualquer-uma
reboot                      # Reinicializa após a instalação. Você provavelmente quererá fazer isto.
bootloader --location=mbr   # Instala o carregador de inicialização padrão na MBR.
install                     # Instala um novo sistema em vez de atualizar.
url --url http://installserver/redhat
clearpart --all --initlabel
part / --fstype ext3 --size 4096
part swap --size 1024
part /var --fstype ext3 --size 1 --grow
network --bootproto dhcp
auth --useshadow --enablemd5
firewall --disabled
xconfig --defaultdesktop=GNOME --startxonboot --resolution 1280x1024 --depth 24
```

As diretivas na seção de comandos especificam uma lista de partições com tamanhos. Você também pode designar uma das partições para crescer até preencher qualquer espaço remanescente no disco.

A opção `rootpw` configura a nova senha root da máquina. O padrão é especificar a senha em texto puro, que pode ser um tanto perigoso em termos de segurança. É possível usar o flag `--iscrypted` para especificar uma senha já criptografada; porém, senhas MD5 não são suportadas.

O Kickstart usa um modo gráfico como padrão, que tolhe o objetivo de instalação sem intervenção do operador. A palavra-chave `text` no início do exemplo corrige isso.

A segunda seção é uma lista dos pacotes a serem instalados, começando com uma diretiva `%packages`. A lista pode conter pacotes individuais, conjuntos como `@ GNOME` ou a notação `@ Everything` para incluir a tralha toda. Ao selecionar pacotes individuais, especifique somente o nome do pacote, não a versão ou a extensão **.rpm**. Eis um exemplo:

www.snow666.blogspot.com
Snow666


```
%packages
@ Networked Workstation
@ X Window System
@ GNOME
meupacotelocal
```

Na terceira seção do arquivo de configuração de Kickstart, é possível especificar comandos de shell arbitrários para Kickstart para serem executados. Há dois possíveis conjuntos de comandos: um introduzido com `%pre` que é executado antes da instalação e um introduzido com `%post` que é executado depois. Ambas as seções possuem algumas restrições na habilidade de o sistema determinar nomes de hosts, de modo que é mais seguro usar endereços IP caso queira acessar a rede. Além disso, os comandos de pós-instalação são executados num ambiente em que foi aplicado **chroot**, de maneira que eles não poderão acessar a mídia de instalação.

Construindo um servidor Kickstart

O Kickstart espera que seus arquivos sejam dispostos da mesma forma como se encontram no CD. Ele quer ter seus pacotes armazenados num diretório chamado **RedHat/RPMS** no servidor. Você pode facilmente adicionar seus próprios pacotes a esse diretório. Há, entretanto, uma série de problemas.

Primeiramente, se você solicitar ao Kickstart para instalar todos os pacotes (com um `@ Everything` na seção de pacotes do arquivo **ks.cfg**), ele irá instalar os pacotes adicionais em ordem alfabética depois de os pacotes básicos terem sido instalados. Se seu pacote depende de outros que não estejam no conjunto básico, você chamaria o seu pacote de algo como **zzmeupacotelocal.rpm** para se certificar que ele seja instalado por último.

Se você quiser instalar todos os pacotes, liste seus pacotes suplementares individualmente na seção `%packages` do arquivo **ks.cfg** ou então adicione seus pacotes a uma ou mais das listas de conjuntos. As listas dos conjuntos são especificadas por entradas como `@ GNOME` e significa um conjunto predefinido de pacotes cujos membros são enumerados no arquivo **RedHat/base/comps** no servidor. Infelizmente, o formato do arquivo **comps** não é bem documentado. Os conjuntos são as linhas que começam com 0 ou 1 (o número especifica se o conjunto é selecionado ou não por padrão). Geralmente, não é uma boa idéia ficar mexendo nos conjuntos-padrão. Sugerimos que você os deixe como a Red Hat os definiu e cite explicitamente todos os seus pacotes suplementares no arquivo **ks.cfg**.

Apontando Kickstart para o seu arquivo de configuração

Assim que tiver criado um arquivo de configuração, há uma série de maneiras de fazer com que Kickstart o utilize. O método sancionado pelo Red Hat é inicializar com um disquete ou CD-ROM de inicialização do Red Hat e solicitar uma instalação Kickstart especificando **linux ks** no prompt inicial `boot:`. Se você não fornecer nenhum argumento adicional, o sistema determinará o seu endereço de rede usando DHCP. Ele obterá então o servidor de inicialização DHCP e as opções do arquivo de inicialização, tentará montar o servidor de inicialização com NFS e usará o valor da opção do arquivo de inicialização como seu arquivo **ks.cfg**. Se nenhum arquivo de inicialização tiver sido especificado, o sistema procurará um arquivo chamado **/kickstart/***hostipaddress***-kickstart**.

Alternativamente, o Kickstart pode ser informado para obter o seu arquivo de configuração de alguma outra maneira fornecendo um caminho como um argumento para a opção **ks**. Há várias possibilidades. A instrução

```
boot: linux ks=http:servidor:/path
```

diz ao Kickstart que use o HTTP para fazer o download do arquivo em vez do NFS. Usar **ks=floppy** diz ao Kickstart que procure **ks.cfg** na unidade de disquete local.

Para eliminar completamente o emprego da mídia de inicialização, você precisará partir para o PXE. Veja a seção 23.2 ara obter maiores informações sobre isso.

YaST: A ferramenta de instalação do SuSE

YaST é a ferramenta de instalação e de configuração do SuSE. A parte referente à instalação é mais ou menos equivalente à combinação Anaconda/Kickstart do Red Hat. Assim como o Kickstart, o YaST opera instalando pacotes RPM; ele também pode ser automatizado com um arquivo de configuração.

O YaST é mais inteligente do que o Kickstart em relação à detecção de hardware e também torna a personalização mais fácil. Ele é menos flexível sob um ponto de vista de inicialização e suporta somente NFS. Um artigo sobre o YaST em

http://www.suse.de/en/whitepapers/yast/index.html

www.snow666.blogspot.com
Snow666

fornece grande parte das informações necessárias para emprego do YaST. Talvez você queira ler o trabalho "Automated Installation of Linux Systems Using YaST" que Dirk Hohndel e Fabian Herschel apresentaram na conferência LISA da USENIX:

http://www.usenix.org/events/lisa99/full_papers/hohndel/hohndel_html

O YaST não suporta PXE na configuração que vem de fábrica, mas fazê-lo funcionar com o PXE é bem mais fácil do que o procedimento equivalente com o Kickstart. Para maiores detalhes, veja as instruções de Anas Nashif para automatizar instalações do YaST em

http://www.suse.de/~nashif/autoinstall/index.html

Criando um servidor YaST

A documentação do SuSE descreve como configurar um servidor YaST baseado em NFS, copiando os CDs do SuSE para o disco rígido e configurando uma árvore de symlinks. O real significado disso é que o YaST procura uma árvore de pacotes e um arquivo de configuração num diretório chamado **suse**. Se você copiá-los, tudo bem. Você também poderá inserir seus próprios pacotes nesse diretório.

Criando uma instalação YaST

A criação de scripts no YaST é por meio de arquivos **info** hierárquicos. O arquivo **info** básico em seu servidor ou CD-ROM YaST pode ser sobreposto por um designado para determinada classe de máquinas. As classes são especificadas por meio de intervalos de endereços IP ou de padrões de nomes de host; por exemplo, *.cs.colorado.edu. Por outro lado, os arquivos de classes podem ser sobrepostos por arquivos específicos de host. Melhor ainda, o YaST pode usar DHCP para determinar parâmetros tais como os endereços IP do host. Caso use DHCP, nem precisará de um arquivo **info** específico para host para configurar os parâmetros da rede.

Oficialmente, o YaST suporta somente inicialização por meio de um disquete ou CD-ROM. Entretanto, você poderá usar PXE com YaST instalando PXELinux e usando o arquivo **initrd** fornecido pelo SuSE no disquete de inicialização como arquivo **initrd** do PXELinux.

Um problema ao usar um disquete de inicialização é que os discos de inicialização padrão do SuSE não são muito inteligentes em relação à configuração de interfaces. Teoricamente, acrescentar um comando **insmod** apropriado ao seu arquivo **info** (como, por exemplo, **insmod 3c59x**) para identificar o número do modelo de sua placa de rede pode ajudar a solucionar este problema. Alternativamente, você pode construir seu próprio kernel para disquete de inicialização incluindo os dispositivos de rede (e de disco) necessários. Veja o Capítulo 12, *Drivers e o Kernel*, para maiores informações sobre a criação de kernels (é fácil!).

Os arquivos **info** residem em **suse/setup/descr/info**, tanto no servidor de instalação quanto no disquete de inicialização. Configure os padrões no arquivo **info** em seu servidor NFS e deixe modificações referentes ao host para o arquivo **info** específico ao host.

Eis um exemplo de um arquivo **info** específico a um host:

```
Language: English
Bootmode: Net                   # Necessário para instalação da rede
Keytable: us
IP: 172.16.10.27
Netmask: 255.255.255.0
Gateway: 172.16.10.1
Server: 172.18.5.9              # Servidor NFS por meio do qual obter pacotes
Serverdir: /susedir
Nameserver: 172.18.5.2
Netdevice: eth0
```

Cada um desses parâmetros pode ser configurado com DHCP. `Server` e `Serverdir` são especificados pelas opções DHCP `bootserver` e `bootpath`.

Também há parâmetros de instalação que especificam o nível de interatividade desejado, mas eles não são consistentes. Alguns usam 2 para significar não interativo, outros usam 1 ou então 0. O pequeno trecho de configuração a seguir ilustra como desativar toda a interatividade:

```
FAST_INSTALL 2      # Supõe que o usuário deseja uma instalação automática.
AUTO_FDISK 2        # 2 significa automaticamente fazer partição no disco.
AUTO_LILO 2         # 2 significa automatizar LILO.
INSTALL_WAIT 0      # Não aguarde por uma entrada de usuário após a instalação de pacotes.
```

www.snow666.blogspot.com
Snow666


```
NO_ASK_SWAP 1       # Não pergunte sobre o swap.
NEVER_STOP 1        # Não pare caso haja algum problema.
```

Para informar o YaST para usar DHCP configurado:

```
AUTO_NAMESERVER 1
AUTO_NAME 1
AUTO_NET 1          # instala automaticamente coisas referentes à rede baseado na instalação
```

O YaST sabe quais pacotes instalar por meio de uma opção `AUTO_INSTALL`. O argumento para essa opção é o caminho para um "arquivo de seleção" que é uma lista dos pacotes a serem instalados. O formato de um arquivo de seleção é basicamente uma lista; adicione os seus pacotes após a linha `Toinstall`. Se você iniciar o caminho com `$I:`, o YaST usará o caminho de instalação. Por exemplo, para instalar a lista de pacotes especificada no arquivo **Minimal.sel** (fornecido pelo SuSE):

```
AUTO_INSTALL $I:suse/setup/descr/Minimal.sel
```

As opções `PRE_INSTALL` e `POST_INSTALL` especificam arquivos de seleção para pacotes a serem instalados antes ou após `AUTO_INSTALL`. As opções `PRE_SCRIPT` e `POST_SCRIPT` informam YaST sobre scripts que devem ser executados antes ou depois da instalação.

O último trecho do arquivo **info** permite que você ative/desative serviços no arquivo **rc.config** do sistema recém-construído; esta é uma área na qual o YaST é significativamente superior ao Kickstart. Cada linha `RC_CONFIG_0` especifica um nome de variável e seu respectivo valor. Todas as linhas `RC_CONFIG_0` serão usadas para construir o arquivo **rc.config**. Veja *Scripts de inicialização no SuSE* na página 26 para maiores informações sobre o **rc.config**.

Por exemplo, para que **named** inicie no momento da inicialização no sistema recém-instalado, acrescente:

```
RC_CONFIG_0 START_NAMED yes
```

no final do arquivo **info**.

SystemImager

SystemImager (www.systemimager.org) é uma ferramenta de fonte aberto que usa uma abordagem diferente daquela de Kickstart ou YaST. Ele instala imagens completas do sistema operacional em vez de pacotes individuais e é mais rápido do que os instaladores baseados em pacotes. Além disso, você pode facilmente fazer mudanças radicais para a maneira que você deseja que os seus sistemas se pareçam. SystemImager pode operar em SSH, que é útil caso esteja instalando Linux num ambiente hostil. Seu suporte ao PXE é de longe o melhor entre os instaladores comuns.

SystemImager é simples e fácil de ser personalizado, mas frágil. Ele não oferece muito suporte para configuração automática da imagem do sistema durante a instalação; você precisa de máquinas relativamente homogêneas para cada imagem de sistema.

Falta ao SystemImager a lógica de configuração de hardware do Kickstart e do YaST. Ter uma imagem para instalar em vários tipos de hardware pode, algumas vezes, levar um bom tempo em termos de ficar realizando ajustes finos. Por exemplo, com a versão atual você tem de fornecer imagens separadas para sistemas com discos SCSI *versus* discos IDE. O SystemImager não entende o conceito de redes que não estão associadas à interface eth0. Ele também não trata de configuração X Windows; entretanto, o suporte da versão 4 do XFree86 de configuração praticamente automática torna isto menos importante. Veja www.xfree86.org para obter maiores informações sobre o XFree86.

Já que é possível ter tantas imagens quanto quisermos, uma opção é construir uma imagem diferente para cada tipo de hardware suportado. Um número excessivo de imagens criará um pesadelo em termos de manutenção, mas para um cluster de máquinas idênticas, o SystemImager é a maneira mais fácil de duplicar exatamente uma instalação.

SystemImager possui um grupo de suporte ativo e em breve se fundirá com o LUI, um instalador originário na IBM, para produzir "uma família completa de aplicativos para gerenciar todas as suas necessidades de construção de sistemas". VA Software, a empresa que produziu originalmente o SystemImager, não o está mais suportando comercialmente, porém uma grande comunidade de fonte aberto ainda o utiliza.

Funcionamento do SystemImager

O SystemImager fornece uma imagem de inicialização SYSLINUX simples por meio da qual você pode inicializar o sistema via disquete ou CD-ROM ou então fazer o download através de PXE. Essa imagem de instalação contém um único script **rc** que obtém e executa um script de instalação do servidor de imagens. O script de instalação cria partições no disco primário e depois executa um segundo **rsync** para copiar a imagem do sistema do servidor para o cliente.

www.snow666.blogspot.com
Snow666

SystemImager realiza muito pouco em termos de localização: ele configura o nome de host e o endereço IP (caso não esteja usando DHCP), executa **lilo**, reinicializa o computador e o deixa com um computador com uma instalação novinha em folha e (espera-se) operacional.

Uma imagem do sistema é uma hierarquia de diretórios no servidor de instalação que é modificável com comandos de manipulação de arquivos padrão. As modificações são mais convenientes do que ocorre com sistemas baseados em pacotes, o que pode exigir que você adicione pacotes ou insira scripts pós-instalação. Infelizmente, esse recurso também torna tentador proceder a localizações apressadas sem documentá-las. Se você não registrar arduamente o processo de criação de imagens, terá poucas esperanças de reproduzi-las quando chegar o momento de atualização.

Caso precise fazer personalizações para máquinas individuais, adicione código ao script de instalação para implementar suas modificações. Uma outra opção é criar os seus próprios scripts **rc** que são executados na primeira vez em que um sistema novo é inicializado. Por exemplo, um dos autores do SystemImager criou programas para suportar tanto uma instalação SCSI quanto uma IDE com base em um único script.

Configurando SystemImager

O ambiente de inicialização do SystemImager é documentado no site a seguir:

http://systemimager.org/manual/html/theactualstep-by-stephowto.html

Os scripts do SystemImager são pequenos scripts Perl ou de shell. Eles são bem documentados.

SystemImager espera que você forneça um sistema de referência para usar como modelo — seu "cliente ouro". Você deve instalar o pacote-cliente de SystemImager no cliente ouro e instalar o pacote de servidor em seu servidor de instalação. Ambos os pacotes se encontram disponíveis no site do SystemImager, www.systemimager.org.

O fato da configuração do cliente ouro ser copiado *ipsis litteris* para máquinas novas não é, na verdade, tão problemático quanto possa parecer à primeira vista. Desde que o kernel do cliente ouro contenha drivers para todo o hardware usado em seu site, as máquinas novas não têm necessariamente de ser idênticas. Na prática, as maiores preocupações se referem a dispositivos de disco (SCSI/IDE) e de rede. O SystemImager espera que a interface de rede seja eth0; para modificar isto, você precisará modificar os scripts de instalação do SystemImager.

*Veja a seção 18.1 para obter maiores informações sobre **rsync**.*

Assim que tiver configurado o cliente ouro, execute o comando **preparecliente**. A seguir, configure o servidor de imagens instalando o pacote **systemimager-server**. SystemImager prefere que o seu servidor DHCP e servidor-imagem sejam a mesma máquina. Use o script **makedhcpserver** para criar um arquivo de controle de servidor DHCP correspondente à sua instalação SystemImager.

Crie a imagem da instalação executando **getimage** no servidor-imagem. Esse script usa **rsync** para instalar a imagem do cliente ouro no servidor e um script de instalação correspondente. São suportados quatro métodos de atribuição de endereços IP: DHCP estático, DHCP dinâmico, endereços IP estático ou "replicantes" (sempre atribui o mesmo endereço IP).

getimage armazena o script de instalação em **/tftpboot/systemimager**. Trata-se de uma opção interessante, já que o script é recuperado com **rsync**, e não com TFTP. Talvez os autores do SystemImager tenham optado pelo fato de os materiais de bootstrap do PXE (que *são* recuperados com TFTP) também residirem em **/tftpboot**; seria melhor manter tudo junto.

Os clientes SystemImager precisam ser adicionados à configuração DHCP (ou terem um disco de inicialização) e ter links apropriados configurados no diretório **tftpboot** para poderem obter suas informações de rede básicas. O comando **addclients** de SystemImager realizará automaticamente essas etapas para você.

Para efetivamente instalar clientes, tudo o que você precisa é entrar em alguma mídia de inicialização (veja os comandos **mkautoinstallcd** e **mkautoinstalldiskette**) ou iniciar uma inicialização PXE.

O SystemImager vem com uma ferramenta, **updateclient**, que atualiza um cliente para atender à versão atual do cliente ouro. Uma outra ferramenta, **pushupdate**, pode ser executada no servidor para fazer com que todos os clientes SystemImager executem **updateclient**. A lista de exceções **rsync** padrão é bem dinâmica e, provavelmente, você terá de expandi-la para alcançar o comportamento desejado.

Consulte a seção 23.4 para uma discussão mais ampla das questões relacionadas às contínuas atualizações.

Personalizando a instalação SystemImager

Se seu servidor de imagens e servidor DHCP forem máquinas distintas, você precisará personalizar o processo de instalação para informar o SystemImager sobre este fato. Crie um arquivo chamado **local.cfg** no disquete de inicialização para configurar os parâmetros que você precisa alterar. O disquete é um disquete FAT padrão, de modo que você pode lê-lo ou gravá-lo montando-o como um volume FAT ou usando **mtools**. O arquivo **local.cfg** é realmente

www.snow666.blogspot.com
Snow666

um script de shell que é executado pelo script **rcS** na hierarquia SystemImager. As variáveis de ambiente são usadas para especificar opções. Por exemplo, para especificar o seu servidor de imagens, coloque a linha

```
export IMAGESERVER=imageserver.toadranch.com
```

no arquivo **local.cfg**.

Obter um arquivo de configuração com PXE atualmente não é suportado. Seria interessante se você pudesse especificar a identidade do servidor de imagens como uma opção DHCP. Em vez disso, você tem de dar uma mexida no script **rcS** na imagem de inicialização (em **/tftpboot/initrd.gz** no servidor de imagem) para corrigir esses problemas.

No servidor também é necessária alguma personalização se, por exemplo, você quiser alterar a maneira que o SystemImager organiza os discos.

23.3 LOCALIZAÇÃO

Instalar um novo computador é somente metade da batalha. É quase inevitável que seja necessário fazer modificações na configuração genérica para cada máquina. Seguir um conjunto específico de diretrizes em suas localizações pode facilitar as coisas para você e seus usuários e permitir a construção segura que suporta milhares de computadores.

Muitos problemas de localização são, na realidade, procedurais em vez de técnicos e provavelmente você tenha de levá-los em consideração antes mesmo de começar a instalar os computadores. Devemos encarar uma série de problemas e as soluções corretas dependerão da natureza de seu site. A sua organização é muito centralizada ou grupos distintos precisam ter algum grau de controle sobre seus computadores? Os seus usuários são tecnicamente experientes ou são neófitos? Que grau de segurança e de privacidade o seu site requer? Muitas dessas questões em escala macro são vistas no Capítulo 29, *Políticas e Diretrizes*.

Você deve considerar em particular como impedir que cada computador de seu site não fique diferente dos demais. Você também tem de estar apto a garantir que seu plano vise a uma fácil recuperação de falhas de sistemas e de erros por parte do administrador do sistema.

É útil imaginar o seu ambiente Linux como um produto de software, um que seja formado pela distribuição escolhida, pelos seus aplicativos locais, suas personalizações administrativas e seja lá que software você use para administração. O segredo para evitar muitos problemas é a aderência a práticas testadas de engenharia de software: use controle de versão, defina um procedimento de garantia de qualidade e documente seus procedimentos e atualizações.

Caso tenha poucos sistemas, alguns dos problemas mais chatos podem ser ignorados. Porém, nunca é muito cedo considerar como você poderia administrar o crescimento. Sua organização pode crescer de forma muito mais tranqüila se seus procedimentos conseguem ser ampliados para tratar da carga adicional.

Automação

Um esquema de localização que requer intervenção manual em cada novo host não será expansível. Apesar das melhores das intenções, os administradores cometem erros e pulam etapas. De acordo com a Lei de Murphy, essas omissões possuem uma maneira de retornar e afligi-lo no momento em que você menos pode suportá-lo.

Os procedimentos de localização são um excelente candidato à automatização. Independentemente de quão enferrujados estejam os seus conhecimentos de programação, um script Perl normalmente se provará menos suscetível a erros do que uma lista de verificação de 8 páginas. A automação é a "mágica" que permite que arranjos de instalações aumentem para centenas ou milhares de hosts — faça as alterações somente uma vez em vez de repeti-las milhares de vezes.

Criando versões de sua configuração

Se sua instalação tiver milhares de computadores e cada um deles tiver uma configuração própria, você gastará a maior parte de seu tempo tentando descobrir por que um computador apresenta determinado problema enquanto outro não. Certamente, a solução é fazer com que todos os computadores tenham a mesma configuração, certo? Não tão rápido, caro colega!

Cada host possui necessidades específicas em termos de atualização, estabilidade e de uptime. Um administrador de sistemas prudente jamais deveria implementar novas versões de software em massa. Em vez disso, as implantações devem ser em estágios de acordo com um plano gradual que acomode as necessidades de outros grupos e dê tempo para a descoberta de problemas enquanto o potencial deles provocarem danos ainda for limitado. Jamais atualize servidores críticos até ter certa confiança nas alterações que está promovendo[1].

1. Correções de segurança é uma possível exceção a essa regra. É importante tapar brechas de segurança tão logo sejam descobertas. Por outro lado, as correções de segurança algumas vezes introduzem *efetivamente* bugs.

www.snow666.blogspot.com
Snow666

Infelizmente, a maioria dos administradores se depara com a situação de ter que dar suporte a várias configurações de software. Porém, há uma grande diferença em administrar *múltiplas* configurações e *inúmeras* configurações. O segredo é manter um controle rígido sobre o primeiro caso para que elas não se transformem sorrateiramente no segundo caso.

Sua principal arma nessa batalha é o sistema de versões. Trate a instalação de seu sistema operacional local como uma versão de software independente com o seu próprio sistema de numeração e procedimentos de testes. Essa abordagem traz muitas vantagens: você pode testar uma nova versão antes de empregá-la nos computadores do ambiente de produção; você pode facilmente retornar os sistemas para uma versão "segura" em caso de problemas e, mais importante ainda, poderá limitar o número de configurações em seu ambiente para um número pequeno e controlável.

Uma versão interna é formada por uma plataforma de sistema operacional, aplicativos, personalizações para o ambiente operacional local e quaisquer correções necessárias para estes componentes. Cada versão deve ser acompanhada por notas de versão (veja *Documentação* na página 542) e por um procedimento apropriado de validação.

Há várias maneiras de implementar releases com versões. Instalações novas podem usar um sistema como o SystemImager que define uma imagem distinta para cada release. Se você usa um instalador baseado em pacotes como o Kickstart ou YaST, poderá simplesmente enumerar os pacotes específicos (e versões de pacotes) que compõem cada release. Sistemas baseados em pacotes funcionam particularmente bem quando as próprias personalizações locais são empacotadas.

Pense cuidadosamente em relação ao número de releases que deseja manter. Quanto maior for o número de releases existentes, maior será o malabarismo que você terá de fazer. Por outro lado, não corrija o que está funcionando. Atualizar sistemas por motivos gratuitos custa tempo e dinheiro e "estar sempre na crista da onda" normalmente significa "sangria desatada".

Considere também os custos e inconvenientes de um sistema de controle de releases altamente formalizado. Esteja preparado para fazer a análise custo/benefício apropriada à sua instalação e para justificar as decisões tomadas. Procedimentos de controle de releases podem tornar sua equipe administrativa menos ágil e retardar sua capacidade em responder rapidamente a emergências. (Se precisar instalar *imediatamente* uma correção de segurança, por exemplo, talvez você não tenha tempo para seguir procedimentos de controle de releases apropriados.) Porém, esses custos normalmente podem ser minimizados e são tipicamente pequenos quando comparados com os benefícios a longo prazo.

Usuários e departamentos rebeldes

O processo de padronizar configurações de software normalmente causa conflitos. Usuários com tendência técnica (e, algumas vezes, departamentos inteiros) podem achar que um sistema de controle de releases centralizado não é capaz de atender adequadamente suas necessidades em termos de configuração ou suas necessidades em termos de ter controle autônomo sobre os computadores que eles usam.

Seu primeiro impulso pode ser o de tentar forçar tais usuários "rebeldes" a aceitar configurações-padrão de modo que minimize o custo e o tempo necessários para suportá-los. Porém, tal abordagem tirânica normalmente acaba criando insatisfação tanto por parte dos usuários quanto dos administradores do sistema. Tenha em mente que os anseios dos usuários rebeldes são em geral perfeitamente legítimos e que é papel dos administradores de sistemas suportá-*los* — ou, pelo menos, refrear-se quanto a tornar mais difícil a vida dos usuários.

A solução mais desejável (mas normalmente não factível) é integrar os usuários rebeldes no processo de definição do conteúdo dos releases. Se eles tiverem um canal formal, institucionalizado e bem conhecido através do qual poderão afetar o conteúdo dos releases-padrão (ou um release próprio com a supervisão do administrador de sistemas), talvez eles não precisem de tanto controle local. Esse canal de comunicação tem de ser fácil de usar e acima de tudo responsivo; se sugestões são acumuladas por seis meses antes de serem integradas, ninguém terá confiança na capacidade de atender às necessidades individuais do sistema.

A alternativa à estratégia de integração é a de trocar suporte por autonomia. Permita que usuários ou grupos rebeldes façam o que eles bem entendam com o entendimento explícito de que eles também devem assumir a responsabilidade de manterem os sistemas personalizados funcionando. Deixe claro que você não estará à disposição para ficar corrigindo problemas, mas tome cuidado com a maneira pela qual você propõe isto. Isso deve ser apresentado como uma conseqüência inevitável e lamentável dos limitados recursos do administrador de sistemas, jamais como um joguinho de poder impertinente do tipo "faça como eu quero, caso contrário eu levo a bola e vou embora pra casa".

www.snow666.blogspot.com
Snow666

Talvez você tenha de tomar algumas medidas para proteger os sistemas sob sua responsabilidade das regiões autônomas selvagens e sem leis de sua organização. Por exemplo, algumas instalações posicionam essas regiões autônomas atrás de um firewall interno de modo que os "automotivados" possam acidentalmente danificarem-se mutuamente, enquanto o restante da rede os ignora graciosamente.

Independentemente do que você fizer, não se deixe abater e sair catando os destroços da criatividade alheia. Não é uma boa opção para emprego de seu precioso tempo e abre um precedente perigoso. Num aperto, talvez você possa oferecer-se para reinstalar a máquina com uma das configurações suportadas pela administração central; deixe para o usuário pesar os custos e benefícios de receber ajuda *versus* preservar um ambiente personalizado.

Testes

É importante testar alterações antes de liberá-las para o mundo. No mínimo, isso significa que você precisa testar suas próprias modificações (como versões novas de arquivos de configuração como **sendmail.cf**); entretanto, você realmente também deve testar o software distribuído pelo seu fornecedor. Certa vez um fornecedor de UNIX famoso (não Linux...ainda) liberou uma correção que, quando aplicada de certa maneira, realizaria um **rm -rf /**. Imagine instalar essa correção em toda a empresa antes de testá-la — suspeitamos que você acabaria tendo de procurar um novo emprego.

Testar é uma questão especialmente pertinente caso você use um serviço como **apt-get** ou o Red Hat Network que oferece uma capacidade de correção automática. Você não iria querer que seus computadores críticos fizessem o download de uma correção defeituosa; portanto, tais sistemas jamais devem ser conectados diretamente a um serviço de atualização. Um mecanismo razoável é manter à mão uma máquina auto-atualizada para emprego como máquina-padrão. Construa periodicamente novas versões baseadas em instantâneos da máquina-padrão, teste-as e depois as libere lentamente para o restante da empresa.

Certifique-se de coagir os usuários a cooperar com o seu regime de testes. Normalmente eles não estarão particularmente ávidos por fazê-lo, porém a participação deles é vital — é a única maneira pela qual todos os componentes que eles usam efetivamente podem ser cobertos. Você deve, obviamente, ter a sua própria lista de verificação de procedimentos para testes, contudo inevitavelmente haverá omissões e lapsos na abrangência dessa lista. Se aquele pequeno, mas crítico, programa aplicativo do qual você se esqueceu não operar junto com a nova **glibc** que você acabou de liberar, você estará diante de um penoso processo de recuperação do sistema.

Certifique-se de que os usuários tenham à disposição uma maneira fácil de relatar bugs. Se preencher relatórios de bugs for um pé no saco, os usuários normalmente ignorarão os problemas e suporão que "alguém irá fazer isto por ele". Veja a seção 29.5 para obter maiores informações sobre sistemas para rastreio de bugs ("tíquete de problemas").

Se sua organização for global, certifique-se de que outras filiais ajudem no processo de testes. A participação internacional é particularmente valiosa em ambientes multilingüísticos. Se, por exemplo, ninguém na unidade norte-americana fala japonês, é melhor você ir para a unidade de Tóquio para testar qualquer coisa que possa vir a afetar o suporte ao kanji. Um número surpreendente de parâmetros de sistema varia com a localização. A unidade nos Estados Unidos testará as alterações na infra-estrutura de impressão com papel A4, ou as unidades fora dos Estados Unidos simplesmente terão uma surpresa?

Fazendo alterações irreversíveis

Erros é um aspecto factual da vida humana. Portanto, é importante ter um controle das alterações que você faz, de modo que se — o horror dos horrores — você realmente cometer um erro, possa facilmente reverter para uma configuração anterior sabidamente funcional. Se sua instalação usa um sistema de releases baseado em versões formais, o retorno a uma situação anterior pode ser tão fácil quanto reinstalar o release anterior. Mas mesmo que você careça de tal infra-estrutura, há maneiras para facilitar as reversões.

A maneira mais simples para você poder voltar atrás é ter sempre à mão um backup de todos os arquivos que você modifica. Por exemplo, talvez você deva copiar **syslog.conf** para **syslog.conf.old** antes de fazer quaisquer alterações. Manter uma cópia de backup também permite que você faça uma análise das diferenças (**diff**) entre as versões atuais e anterior de um arquivo de configuração caso comece a perder o controle das alterações que você fez.

Se você usar esse sistema de backup manual, adote uma convenção de atribuição de nomes para arquivos de backup e tente fazer com que todos os administradores em sua instalação a utilizem de forma consistente. Algumas instalações usam sinais de subtração (por exemplo, **syslog.conf-**), outras usam um sufixo **.old** ou **.bak**, ao passo que outras usam esquemas de numeração multiversão.

Snow666

Os sistemas regularmente copiados para fitas de backup ainda podem se beneficiar do uso de arquivos de backup criados manualmente. A recuperação de um arquivo de backup é mais rápida e mais fácil do que a recuperação por meio de uma fita, e os backups manuais preservam uma camada adicional de histórico.

Backups manuais funcionam até um certo ponto, mas eles podem rapidamente se tornar um transtorno caso você esteja fazendo uma série de alterações (você preserva várias cópias com nomes diferentes? Ou apenas preserva a mais recente?). Felizmente, a comunidade dos programadores há muito tempo desenvolveu uma solução para esse problema: controle de revisão.

Controle de revisão

Se várias pessoas quiserem editar um arquivo, devem surgir dois problemas em potencial a serem suplantados. Em primeiro lugar, deve haver alguma maneira de nos certificarmos de que não mais de uma pessoa edite o mesmo arquivo ao mesmo tempo; edições simultâneas podem resultar numa condição de disputa que fazem com que algumas alterações editadas sejam permanentemente perdidas.[2] Em segundo lugar, deve haver alguma maneira organizada de rastrear o histórico das modificações de tal forma que as alterações possam ser compreendidas segundo o contexto e de modo que versões anteriores do arquivo não sejam perdidas para sempre. Para solucionar esses problemas, os programadores desenvolveram sistemas de controle de revisão.

Nos velhos tempos do UNIX, o sistema de controle de revisão original chamava-se SCCS. Ele funcionava suficientemente bem, porém custava dinheiro; o RCS (Revision Control System, sistema de controle de revisão) foi desenvolvido por Walter Tichy na forma de uma alternativa em código aberto. Hoje em dia, o RCS se tornou um padrão de fato para UNIX e Linux e o SCCS está muito mais para nota de rodapé histórica. Cada distribuição Linux vem com o RCS e ele é simples e fácil de ser usado.

Estão disponíveis também vários sistemas comerciais para controle de revisão. Se você trabalha numa software house, talvez já tenha um deles disponível. Entretanto, baseado em nossa experiência, os sistemas comerciais geralmente são exagerados para emprego na administração de sistemas.

Uma outra opção é um sistema de fonte aberto chamado CVS (Concurrent Versions System, sistema de versões concorrentes) que acrescenta alguma funcionalidade ao RCS. Ele suporta um modelo distribuído (para uso com um servidor remoto) e melhor suporte multidesenvolvimento. Uma série de instalações tem usado o CVS para tarefas administrativas; em particular, a capacidade cliente/servidor pode ser bem útil. Mas se você estiver partindo da estaca zero, recomendamos começar com o RCS.

RCS

RCS (Revision Control System, sistema de controle de revisões) é um dos mais antigos aplicativos UNIX existentes. Ele é, na verdade, um sistema relativamente simples. Ele opera num nível de arquivos individuais (ao contrário do CVS que é capaz de operar em árvores) e armazena o histórico de revisão de cada arquivo num arquivo espelhado distinto. O nome do arquivo espelhado é o mesmo do original, porém com os caracteres **,v** anexados a ele. Se, por exemplo, você colocou **/etc/syslog.conf** sob controle RCS, o RCS manterá suas revisões em **/etc/syslog.conf,v**.

Para reduzir a bagunça, o RCS procura um diretório chamado **RCS** no mesmo diretório do arquivo original. Se ele existir, o RCS colocará o arquivo **,v** ali em vez de deixá-lo misturado com os outros arquivos. As listagens de diretórios se tornam muito mais limpas dessa maneira, pois muitos arquivos podem compartilhar um diretório **RCS**. Trata-se de um recurso extraordinário e que recomendamos enfaticamente.

O conceito básico do RCS é que você faça o "check out" (registre a saída) dos arquivos que pretende modificar e faça o "check in" (registre a entrada) das suas alterações para armazená-las. Os únicos comandos RCS que você realmente precisa conhecer são **ci (check in)** para registrar a entrada, **co (check out)** para registrar a saída para modificar e **rcs** que realiza várias tarefas de limpeza e manutenção.

Para começar a manter um arquivo sob RCS, execute **ci** nele como se segue:

```
% ci -u syslog.conf
RCS/syslog.conf,v <-- syslog.conf
enter description, terminated with single '.' or end of file:
NOTE: This is NOT the log message!
>> Este é o arquivo de configuração syslog.
```

2. Suponha, por exemplo, que os administradores de sistemas Alice e Roberto tenham editado o mesmo arquivo e que cada um deles faça algumas alterações. Alice salva o arquivo primeiro. Quando Roberto salva sua cópia do arquivo, ele sobregravará a versão de Alice. Se Alice tiver saído do editor, as suas alterações serão completamente perdidas e irrecuperáveis.

www.snow666.blogspot.com
Snow666

A opção **-u** faz com que **ci** registre (a saída) imediatamente o arquivo **syslog.conf** num estado desbloqueado (não editável). Se você omitir esse flag, **ci** registrará (a entrada) o arquivo e eliminará a cópia original, que provavelmente não é o que que você deseja.

Você poderia aplicar **chmod** ao arquivo **syslog.conf** para ser gravável e modificá-lo desta maneira, porém este procedimento iria subverter e confundir o RCS. Toda vez que quiser alterar um arquivo controlado por RCS, você terá de registrar a sua saída e bloqueá-lo com **co -l**:

```
% co -l syslog.conf
RCS/syslog.conf,v --> syslog.conf revision 1.2 (locked)
done
```

Essa operação informa o RCS de que você está prestes a modificar o arquivo; o RCS não deixará mais ninguém retirar o arquivo até que você o tenha devolvido.

Teoricamente, o bloqueio impede duas pessoas diferentes de modificarem o arquivo ao mesmo tempo. Na prática, você tem de ser root para modificar arquivos de sistema, de modo que qualquer um com privilégios **sudo** poderá modificar um arquivo assim que ele for reservado como root. Porém, se um segundo administrador tentar um outro **co -l**, o RCS perceberá que já existe uma versão gravável. Os administradores de sistemas devem se acostumar a sempre tentar reservar os arquivos controlados por RCS que eles queiram modificar. O fato de um arquivo ser gravável significa: "Pare! Alguma outra pessoa já está com este arquivo reservado".

Assim que estiver satisfeito com suas modificações para um arquivo reservado, devolva-o via **ci -u**. O sistema irá lhe solicitar um comentário descrevendo o que você acabou de fazer. Embora seja tentador pular essa etapa ou gravar algo inútil como "feito uma alteração", resista à tentação. Daqui a alguns anos quando estiver tentando descobrir porque você fez algo, esses úteis comentários poderão salvar a sua vida.

```
% ci -u syslog.conf
RCS/syslog.conf,v <-- syslog.conf
new revision: 1.3; previous revision: 1.2
enter log message, terminated with single '.' or end of file:
>> Started logging debug messages to track down problem
>> ^D
done
```

Você pode analisar o histórico de revisão de um arquivo através do comando **rlog**:

```
$ rlog syslog.conf
RCS file: RCS/syslog.conf,v
Working file: syslog.conf
head: 1.3
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 3; selected revisions: 3
description:
----------------------
revision 1.3
date: 2002/01/10 00:44:58; author: adam; state: Exp; lines: +1 -0
Started logging debug messages to track down problem
----------------------
revision 1.2
date: 2000/07/19 08:23:10; author: evi; state: Exp; lines: +2 -0
Changed log destination to new logmaster
----------------------
revision 1.1
date: 1998/03/14 11:13:00; author: matthew; state: Exp;
Initial revision
=============================================================
```

Caso queira verificar o aspecto do arquivo antes de alterar para o novo log master, você poderá registrar a saída da revisão 1.2 do arquivo através da opção **-r** do comando **-co**:

```
% co -r1.2 syslog.conf
```

Snow666


```
RCS/syslog.conf,v --> syslog.conf
revision 1.2
done
```

Esse comando substitui o arquivo **syslog.conf** atual pela versão mais antiga; portanto, certifique-se de fazer regularmente um **co** quando tiver terminado; caso contrário, você (e **syslogd**) poderão ficar muito confusos! Jamais registre a saída de cópias bloqueadas de revisões mais antigas (via **co -l**), já que esta operação criará ramificações na árvore de versões. As ramificações de versão são ocasionalmente úteis para código-fonte, mas jamais são usadas para fins administrativos; simplesmente ignore toda a documentação RCS que lida com elas para simplificar a sua vida.

Ocasionalmente você poderá descobrir que alguém modificou um arquivo e o deixou bloqueado — ou até pior, sobrepôs RCS e modificou o arquivo sem bloqueá-lo. Você poderá rever as alterações feitas pelo perpetrador com **rcsdiff**, que é uma versão de **diff** que reconhece RCS:

```
% rcsdiff syslog.conf
===================================================================
RCS file: RCS/syslog.conf,v
retrieving revision 1.3
diff -r1.3 syslog.conf 4c4
< define(LOGHOST,moonbase)
---
> define(LOGHOST,spacelounge)
%
```

Como último recurso, você poderá desfazer o bloqueio através do comando **rcs -u** *nome_do_arquivo*. Esse comando lhe solicita a introdução de uma explicação sobre seus atos e envia um e-mail para o usuário que havia anteriormente bloqueado o arquivo.

Informações adicionais sobre o RCS podem ser encontradas em

http://www.cs.purdue.edu/homes/trinkle/RCS/

Para informações sobre o CVS, consulte www.cvshome.org.

Documentação

A documentação apropriada das configurações de software é absolutamente crucial para o perfeito funcionamento de sua instalação. Se você fez releases formais de sua configuração local, é vital fornecer notas de versão úteis com cada release novo. Mesmo que seu sistema seja muito pequeno para merecer releases formais, é importante que outros administradores e a sua comunidade de usuários saibam o que você modificou. Você não irá querer que alguém de uma outra unidade telefone às 4 da manhã, pois eles não podem descobrir por que o software deles não funciona no release atual.

Além disso, é útil manter um documento-sumário que descreva em que aspectos a sua instalação difere da distribuição-padrão. Esse documento pode ser mantido de maneira mais útil na forma de uma lista de verificação de localização descrevendo como configurar o seu ambiente local (isto é, que alterações devem ser feitas na instalação básica) e como instalar um novo computador (caso os dois procedimentos difiram, o que normalmente acontece).

Questões referentes à documentação são vistas de forma mais detalhada no Capítulo 29, *Políticas e Diretrizes*.

Onde instalar software local

Nos velhos tempos do UNIX, quando havia muitas arquiteturas diferentes, programas eram geralmente distribuídos na forma de arquivos-fonte, normalmente arquivos **.tar.Z** que seriam descompactados (**uncompress**) e depois compilados. Assim que o programa fosse construído, seria instalado o software numa localização como **/usr/local**. Hoje o emprego de sistemas de gerenciamento de pacotes significa que um número menor de programas precisa ser instalado dessa maneira. Isso também significa que os administradores tomam menos decisões, já que os pacotes especificam onde seus conteúdos devem ser instalados.

Até mesmo com esse fácil gerenciamento de pacotes, muitas pessoas ainda preferirão compilar seu próprio software. Executar a versão que você mesmo compilou lhe dá um maior controle sobre as opções compiladas no software. Ela também lhe permite ser mais paranóico, pois você pode inspecionar o código-fonte que você está compilando. Algumas pessoas pensam que isso é importante, mas a menos que você tenha o tempo e conhecimento para inspecionar cada linha de um pacote de software de 20.000 linhas, suspeitamos que o valor de segurança agregado é provavelmente mínimo.

www.snow666.blogspot.com
Snow666

Já que nem todo software no mundo foi empacotado para todas as distribuições Linux, é bem provável que você irá se deparar com pelo menos alguns programas em que terá de fazer a instalação, especialmente se os seus computadores Linux não forem baseados em IA-32. Mais ainda, se a sua instalação tem como objetivo o desenvolvimento de sistemas, você terá de considerar onde colocar o seu próprio software desenvolvido localmente para a instalação.

Historicamente, a localização mais comum para software local tem sido **/usr/local** e esta convenção ainda é amplamente seguida hoje em dia. O padrão FHS (Filesystem Hierarchy Standard, padrão de hierarquia de sistema de arquivos) do UNIX/Linux especifica que **/usr/local** esteja presente e vazio após a instalação inicial do sistema operacional e muitos pacotes esperam ser instalados neste local. Um número cada vez maior de outros pacotes (particularmente aplicativos comerciais) esperam ser instalados em **/usr**, o que é uma má idéia e deve ser evitado; até onde você puder, mantenha **/usr** mais ou menos como sua distribuição o instalou. Caso tenha algum aplicativo que insista em colocar arquivos em **/usr**, sugerimos que você efetivamente posicione os arquivos em algum outro lugar e simplesmente coloque symlinks para eles em **/usr**.

Embora **/usr/local** seja tradicional, muitas instalações o classificam como um terreno inapropriado. A maneira tradicional que é disposta (basicamente a mesma que **/usr**, com binários em **/usr/local/bin**, páginas de manual em **/usr/local/man** e assim por diante) cria um montão de problemas: é difícil ter várias versões instaladas do mesmo software, os diretórios podem ficar grandes, é problemático gerenciar várias arquiteturas etc.

Uma ferramenta útil para manutenção de hierarquias é o Stow; veja em www.gnu.org.

Sites maiores normalmente se reorganizam para criar suas próprias convenções de espaço de nomes. Esse espaço de nomes é normalmente montado através de um sistema de arquivos de rede como o NFS ou o Coda (as capacidades de ambos não normalmente usadas para suportar várias arquiteturas). Uma técnica comum é utilizar uma hierarquia de três camadas com um fornecedor ou conjunto na camada mais alta, um nível de aplicação abaixo desta e uma camada de versão abaixo desta. O agrupamento de camada mais alta pode se basear em fornecedor, em função ou, para aplicativos internos, no grupo que desenvolveu o produto. Se, por exemplo, você agrupar por função, pode haver um diretório **graphics** contendo **gimp**, **xv** e outras ferramentas gráficas.

```
/tools/graphics/bin/gimp (symlink to ../gimp/1.2.3/bin/gimp)
/tools/graphics/gimp/1.04
/tools/graphics/gimp/1.2.3
/tools/graphics/xv/3.1.0/
/tools/editors/xemacs/21.4
```

A seguir, você poderia criar um diretório **/tools/graphics/bin** que contivesse symlinks para os binários mais atuais para cada pacote. Se, por exemplo, alguém precisasse usar **gimp** 1.0.4, ele poderia executá-lo diretamente. Uma das desvantagens dessa abordagem é que as "ferramentas antigas jamais morrem". Considere a possibilidade de anunciar cronogramas de "final de vida" para versões antigas dos pacotes assim que tornar disponível uma versão mais nova.

Quanto instalar nas máquinas-cliente

Onde o software local deveria ser realmente instalado: em clientes individuais ou num servidor de arquivos central por meio do qual ele poderia ser compartilhado via NFS? A solução NFS faz atualizações de forma muito mais rápida (é mais rápido atualizar 10 servidores NFS do que 1.000 clientes), salva espaço em disco nos clientes (embora no mundo dos discos de 50 GB, isto não pareça ser mais problema) e permite backups mais fáceis (se houver alguma situação local que exija, é mais fácil fazer o backup de 10 servidores do que 1.000 clientes).

A questão realmente se resume na maneabilidade **versus** velocidade e confiança. O acesso remoto é centralizado e mais fácil de ser controlado, mas acessar a rede sempre será mais lento do que acessar um disco local. Além disso, o modelo de servidor remoto adiciona dependências da rede e do servidor de arquivos remotos.

Estações de trabalho sem discos ("diskless") eram comuns nos velhos tempos do UNIX quando os discos rígidos eram muito caros. Essas máquinas não tinham nenhum disco e obtinham todos os seus arquivos (inclusive suas partições-raiz e espaço de swap) via rede. Não recomendamos essa configuração (as estações de trabalho sem discos morreram por uma razão), mas o FHS realmente permite que todo o **/usr** seja compartilhado remotamente e muitas instalações tiram proveito deste recurso.

23.4 Mantendo Seus Sistemas Atualizados com rsync ou rdist

OK, você se converteu à religião "release centralizado" e definiu algumas configurações específicas que está querendo suportar. Como agora transformar cada um dos computadores como os mestres e como atualizá-los no futuro?

www.snow666.blogspot.com
Snow666

Veja o Capítulo 18, para uma revisão de algumas das maneiras de manter sistemas de arquivos sincronizados.

Há duas maneiras comuns: copiar arquivos diretamente de um host mestre para seus computadores individuais usando uma ferramenta como o **rsync** ou usar o sistema de gerenciamento de pacotes embutido em sua distribuição para instalar pacotes novos ou atualizados. A última opção pode ser automatizada com um sistema como **apt-get**.

Algumas instalações fazem atualizações segundo um cronograma regular; este plano define um limite superior nos sistemas-cliente desatualizados. Outras instalações atualizam seus computadores no momento da inicialização; este planejamento é seguro, mas pode significar que se passa um longo período entre atualizações. Algumas instalações com usuários tecnicamente experientes deixam que os próprios usuários escolham quando atualizar suas máquinas.

Trata-se de um plano cooperativo e não intrusivo, mas você poderá descobrir que alguns usuários jamais atualizam sem serem incitados. O emprego de **rsync** ou **rdist** é de longe a técnica mais antiga e tradicional para manter os sistemas sincronizados. No servidor, basta definir uma tarefa **cron** a cada noite (ou de hora em hora, ou semanalmente...) que aplique **rsyncs** ou **rdists** às máquinas-cliente. Com um sistema desses, você garante que as máquinas sejam mantidas atualizadas desde que estejam ligadas.

Como alternativa, talvez você queira considerar o emprego de um mecanismo de "extração" no qual os clientes são responsáveis por solicitarem a atualização. Um sistema de extração dá aos clientes melhor controle sobre seus próprios cronogramas de atualização. Por exemplo, numa instalação que faz atualizações "todas as noites", o horário de atualização apropriado variará dependendo do fuso horário; uma atualização a cada noite nos Estados Unidos corresponderia a uma atualização no meio de uma jornada de trabalho na Ásia. Você pode facilmente implementar extrações de atualizações executando **rsync** em modo daemon no servidor e fazendo cada um dos clientes disparar o seu próprio serviço **rsync**. SystemImager também facilita extrair atualizações através da ferramenta **updateclient**.

rsync geralmente é um protocolo melhor do que **rdist**, porém ambos são fáceis de serem configurados; veja a página 367 para detalhes. Dependendo do número de máquinas e qual a extensão da área geográfica coberta por elas, talvez você queira configurar somente um servidor de distribuição ou então uma hierarquia de servidores de distribuição. Você poderia, por exemplo, ter um servidor mestre central que faz a distribuição para um escravo em cada um dos prédios nos quais você tem máquinas e que, por sua vez, estes escravos fariam a distribuição para os clientes. Esse tipo de arranjo hierárquico pode efetivamente reduzir o consumo de largura de banda WAN.

Usar **rsync/rdist** para manter atualizado um sistema de arquivos como **/tools** é bem objetivo; há poucas pegadinhas ocultas. Entretanto, é mais difícil usar essas ferramentas para atualizar suas instalações de sistema operacional centrais. Primeiramente você precisará manter uma lista dos arquivos que não devem ser tocados durante a atualização. Essa lista deve incluir arquivos que definem informações pessoais de um host (como **/etc/sysconfig/network** no Red Hat), arquivos criados por daemons (como quase tudo que termina em **.pid**), arquivos temporários (**/tmp** e **/var/tmp**) e arquivos relacionados a processos em execução (como **/proc**). A maior parte de **/var** não deve ser mexida. Dependendo do nível de consistência que você quer instaurar, talvez vá querer ou não que algumas partes de **/etc** sejam copiadas.

Procedimentos especiais têm de ser usados para acomodar as atualizações de certos arquivos. Por exemplo, você deve executar **lilo** novamente ao instalar um novo kernel. Esse é na verdade o aspecto pelo qual **rdist** é melhor do que **rsync**: ele possui uma ação `special` que acomoda estas considerações. Entretanto, o mesmo efeito pode ser obtido via **rsync** com alguns ajustes adicionais. Talvez você queira executar um script de shell após o **rsync** que faz a limpeza ou então adicionar um serviço **cron** que execute **lilo** toda noite.

A seguir temos um exemplo de lista de exclusões que impedirão uma máquina Red Hat de ser metralhada por um **rsync**. É quase certo que você terá de adicionar outros arquivos que não queira que sejam copiados (por exemplo, **/etc/resolv.conf** caso não queira compartilhá-los em todos os pontos), mas a lista a seguir é um bom ponto de partida:

```
/proc/*
/var/lock/*
/var/log/*
/var/run/*
/var/spool/*
/dev/*
/var/adm/*
/var/tmp/*
/tmp/*
/etc/sysconfig/*
```

Snow666


```
/etc/modules.conf/*
/etc/HOSTNAME
/etc/X11/XF86Config
```

Digamos que você armazene essa lista em **/etc/rsync.excludes**. Para copiar os arquivos de sua máquina para o host desthost, você poderia executar o seguinte comando:

```
# rsync --exclude-from=/etc/rsync.excludes / desthost:/
```

Uma outra grande questão em manter o sistema operacional central atualizado é como controlar os pacotes. Uma maneira de fazer isso (ilustrada no arquivo de exclusões acima) é simplesmente copiar os dados de pacotes (**/var/lib/rpm** em sistemas baseados em RPM, **/var/lib/dpkg** no Debian) juntamente com o restante do sistema operacional. Isso significa que na prática seus clientes *não estarão* aptos a adicionar pacotes próprios; se eles fizerem, os dados do pacote serão sobregravados. Vistos sob um determinado ângulo, isso poderia na verdade ser um recurso — se os clientes possuem pacotes adicionados a eles, estes não estão padronizados o que aumenta a carga de administração e provoca problemas com reprodutibilidade. Caso sinta que seja útil permitir que clientes possuam conjuntos diferentes de pacotes instalados, provavelmente você irá querer controlar as atualizações por meio de pacotes.

23.5 Administração de Pacotes

Nos dias atuais, a maioria das distribuições Linux (incluindo todas aquelas apresentadas como exemplo neste livro) usa algum tipo de sistema de empacotamento. Os pacotes são usados para distribuir aplicativos e algumas vezes arquivos de configuração. Eles possuem várias vantagens em relação aos tradicionais arquivos não estruturados **.tar.gz**. Talvez o mais importante seja que eles tentam tornar o processo de instalação o mais infinitesimal possível. Se ocorrer algum erro, o pacote poderá ser revertido ou reaplicado.

Instaladores de pacotes também fazem backup dos arquivos que eles alteram. Caso você descubra que algum pacote recém-instalado esteja causando problemas no sistema, você pode, teoricamente, voltar atrás para restaurar o seu sistema ao seu estado original.

Os sistemas de empacotamento definem um modelo de dependência que permite aos mantenedores de pacotes garantirem que todas as bibliotecas e infra-estrutura de suporte das quais seus aplicativos dependem sejam instaladas apropriadamente. Os pacotes também podem executar scripts em vários momentos durante a instalação, de modo que eles podem fazer mais do que simplesmente instalar arquivos novos. (Esse recurso provavelmente é responsável pela maioria das falhas observadas nos pacotes em restaurar o sistema ao seu estado original após uma desinstalação.)

Há dois formatos de pacotes sendo usados comumente. SuSE, Red Hat e outras distribuições usam RPM, o Package Manager do Red Hat. O Debian usa pacotes no formato **.deb**. Os dois formatos de pacotes são funcionalmente idênticos e, portanto, batalhas enormes têm sido travadas para dizer quem é o melhor. Na prática, o RPM venceu a guerra; porém, é melhor você aderir ao mecanismo de pacotes nativos usados por sua distribuição. É mais fácil fazer a conversão entre os dois formatos de pacotes com uma ferramenta como o Alien no endereço

http://kitenet.net/programs/alien

Os pacotes podem ser uma maneira excelente de distribuir suas próprias localizações devido a todas as vantagens que eles oferecem em relação aos arquivos **.tar.gz**. Você pode criar facilmente um pacote que, quando instalado, lê informações de localização sobre uma máquina (ou as obtêm de um banco de dados centralizado) e usa essas informações para definir arquivos de configuração. Também é possível juntar seus aplicativos locais como pacotes, completar com dependências e criar pacotes por meio de aplicativos de terceiros que não são normalmente distribuídos como pacotes. Você pode criar versões de seus pacotes e usar o mecanismo de dependência para atualizar a máquina quando a nova versão de seu pacote de localização for instalado. É possível, por exemplo, criar um pacote que não instala nada próprio porém dependa de várias outras correções.

Instalar o pacote com dependências ativadas resultará na instalação de todas as correções. Criar pacotes novos é relativamente fácil e alguns excelentes documentos descrevem o processo. Para maiores informações sobre pacotes RPM, veja www.rpm.org ou o livro *Maximum RPM* de Ed Bailey.

Para informações sobre como criar pacotes **.deb** estilo Debian, veja o *Debian New Maintainers' Guide* em

http://www.debian.org/doc/maint-guide

Snow666

Ele contém instruções para construção de pacotes juntamente com diretrizes úteis para torná-los gerenciáveis. Informações genéricas sobre o formato dos pacotes **.deb** podem ser encontradas no Capítulo 6 do Debian FAQ em

http://www.debian.org/doc/FAQ/ch-pkg_basics.html

Pacotes RPM

Controlamos pacotes RPM através do comando **rpm**. Esse único comando faz praticamente tudo o que se relaciona a RPMs, inclusive sua construção. **rpm --help** apresenta uma lista de todas as opções disponíveis.

Caso não esteja construindo pacotes, as únicas opções para as quais você deve prestar atenção são **--install**, **--upgrade** e **--erase**; a opção **--query** determina o estado e o conteúdo dos pacotes e ela também é indispensável. A opção **--query** na realidade somente coloca **rpm** no modo de consulta. Você deve fornecer uma opção adicional que resolva a sua questão específica. O comando **rpm --query --all** mostra uma lista completa dos pacotes instalados no sistema.

Suponha que você precise instalar uma nova versão de OpenSSH devido a uma correção de segurança que foi recentemente publicada. Em primeiro lugar, você precisa localizar o pacote atualizado. Se o pacote fizer parte do sistema operacional central, muito provavelmente haverá uma cópia no site FTP de sua distribuição; o relatório de segurança o informará onde obtê-lo. Se o pacote for mais obscuro, seu primeiro método de pesquisa deve ser verificar rpmfind.net. Esse site Web fornece um interessante mecanismo de busca RPM que o informa exatamente onde podem ser encontradas as diversas versões de um RPM para várias distribuições.

Assim que tiver feito o download do pacote para o seu computador local, instalá-lo requer apenas um simples **rpm --upgrade**:

```
# rpm --upgrade openssh-2.9p2-12.i386.rpm
error: failed dependencies:
openssh = 2.9p2-7 is needed by openssh-askpass-2.9p2-7
openssh = 2.9p2-7 is needed by openssh-askpass-gnome-2.9p2-7
openssh = 2.9p2-7 is needed by openssh-clients-2.9p2-7
openssh = 2.9p2-7 is needed by openssh-server-2.9p2-7
```

Opa! Bem, talvez não seja tão simples assim. Aqui observamos que a versão atualmente instalada de OpenSSH, 2.9p2-7, é exigida por uma série de outros pacotes. O comando **rpm** não permitirá atualizarmos o OpenSSH, pois a mudança poderá afetar todos os demais pacotes. Poderíamos forçar a atualização com a opção **--force**, mas normalmente esta não é uma boa idéia. Em vez disso, capturaremos o restante dos pacotes. (O aviso de segurança original que usamos para esse exemplo listava, na verdade, os demais pacotes — observe o poder da documentação!)

Também poderíamos ter determinado que outros pacotes dependeriam de OpenSSH da seguinte maneira:

```
# rpm --query --whatrequires openssh
openssh-askpass-2.9p2-7
openssh-askpass-gnome-2.9p2-7
openssh-clients-2.9p2-7
openssh-server-2.9p2-7
```

Suponha que tivéssemos obtido cópias atualizadas de todos os pacotes. Poderíamos instalá-los um de cada vez, porém **rpm** é inteligente o suficiente para podermos instalá-los de uma só vez.[3] Se listarmos vários RPMs na linha de comandos, **rpm** os colocará em ordem segundo a dependência antes de instalá-los.

```
# rpm --upgrade openssh-*
```

Legal! Parece que deu certo, e com certeza:

```
# rpm --query openssh
openssh-2.9p2-12
```

Pacotes Debian

Da mesma forma que os pacotes RPM possuem o comando **rpm**, os pacotes Debian possuem o comando **dpkg**. **dpkg** faz praticamente tudo o que você, como usuário final, precisa fazer com pacotes Debian, inclusive os instala (a opção **--install**) e os remove (a opção **--remove**). Podemos listar os pacotes existentes no sistema com **dpkg -l**.

3. Na verdade, versões anteriores de **rpm** requeriam que os especificássemos na ordem correta.

Snow666

Suponha que a equipe de segurança do Debian tenha lançado recentemente uma correção para **nvi** para corrigir um problema de segurança em potencial. Após pegar a correção, executamos **dpkg** para instalá-la. Como pode ser observado, é muito mais loquaz do que **rpm** e nos informa exatamente o que está fazendo.

```
# dpkg --install ./nvi_1.79-16a.1_i386.deb
(Reading database ... 24368 files and directories currently installed.)
Preparing to replace nvi 1.79-14 (using ./nvi_1.79-16a.1_i386.deb) ...
Unpacking replacement nvi ...
Setting up nvi (1.79-16a.1) ...
Checking available versions of ex, updating links in /etc/alternatives ...
(You may modify the symlinks there yourself if desired - see 'man ln'.)
Leaving ex (/usr/bin/ex) pointing to /usr/bin/nex.
Leaving ex.1.gz (/usr/share/man/man1/ex.1.gz) pointing to
   /usr/share/man/man1/nex.1.gz.
...
```

Agora podemos usar **dpkg -l** para verificar se a instalação funcionou. O flag **-l** aceita um padrão de pesquisa opcional, de maneira que podemos somente procurar **nvi**:

```
% dpkg -l nvi
Desired=Unknown/Install/Remove/Purge
| Status=Not/Installed/Config-files/Unpacked/Failed-config/Half-installed
| / Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
| | / Name       Version        Description
+++-==========-==============-================================
i I   nvi        1.79-16a.1     4.4BSD re-implementation of vi.
```

Parece que nossa instalação funcionou perfeitamente.

Automatizando a instalação de pacotes

Vários programas permitem que atualizemos pacotes de modo automático. Os dois programas mais refinados e populares destes são o Red Hat Network e o **apt-get** do Debian.

O Red Hat Network está intimamente acoplado à distribuição Red Hat. Trata-se de um serviço comercial que custa caro e oferece muito mais em termos de atraentes interfaces gráficas com o usuário e de capacidade de automação do que o **apt-get**. Infelizmente, se trata de uma misteriosa caixa-preta oculta.

apt-get é mais bem documentado do que o Red Hat Network, é significativamente mais portável e é gratuito. É também mais flexível em termos do que você pode fazer com ele. **apt-get** é mais firmemente integrado ao Debian e **dpkg**, mas há uma versão que usa RPMs e funciona com o Red Hat e o SuSE. Sugerimos que você use essa versão de **apt-get** caso queira configurar sua própria rede de distribuição de pacotes automatizada. Veja o tópico *Configurando um servidor APT interno*, na seção 23.6, para obter maiores informações.

Além dos extratores de pacotes automatizados, há algumas ferramentas que não são especificamente orientadas a pacotes mas que permitem a execução em máquinas-cliente de tarefas relacionadas a pacotes. Uma das mais conhecidas é o **cfengine**, que é uma ferramenta que lhe permite executar tarefas em vários hosts. Ela é útil para um bom número de tarefas administrativas, não somente na instalação de pacotes. Veja

http://www.iu.hio.no/cfengine

para obter maiores informações.

Red Hat Network

Poderíamos descrever o Red Hat Network como uma tentativa para você na verdade comprar a versão oficial do Red Hat Linux em vez de fazer o seu download na Internet. Paga-se uma taxa de assinatura mensal ou anual e, em troca, você se mantém atualizado em relação à árvore mestra do Red Hat. Já que o "aluguel de software" parece estar na moda atualmente, é bom ver a Red Hat trazendo o Linux para a crista da onda dos modelos de carregamento de software.

Em sua acepção mais simples, pode-se simplesmente usar o Red Hat Network como um portal Web glorificado e lista de discussão. Usá-lo dessa forma não é muito diferente das listas de discussão para notificação de correções que têm sido usadas há anos por vários fabricantes de UNIX. Mas há mais recursos disponíveis caso esteja pronto para pagar por eles.

O Red Hat Network dispõe de uma bela interface gráfica com o usuário para fazer o download de pacotes novos (há também uma alternativa em linha de comando). Ele até permite que você faça o download e instale pacotes

Snow666

novos sem qualquer intervenção humana. Assim que tiver feito o registro, suas máquinas poderão pegar todas as correções para bugs sem jamais você ter de abandonar a sua partida de Quake. O inconveniente do registro automático é o fato de o Red Hat decidir de quais atualizações você precisa. Talvez você possa considerar o quanto você confia no Red Hat (e nos mantenedores de software cujos produtos eles englobam) em não bagunçar as coisas. Dado algumas das interessantes opções oferecidas pela Red Hat no passado no que se refere a pequenas coisas como qual compilador oferecer, algumas pessoas podem permanecer cépticas. Um compromisso razoável poderia ser fazer a assinatura de atualizações automáticas para uma máquina de sua empresa. Você poderá então extrair instantâneos daquela máquina em intervalos periódicos para testar como possíveis candidatos para releases internos.

Para registro do Red Hat Network, execute o programa **rhn_register** que acompanha o Red Hat ou então use a interface Web em rhn.redhat.com. Se estiver protegido por um firewall, execute **rhn_register --configure** para configurar um proxy; caso contrário, use a interface gráfica do Red Hat para fazê-lo.

Assim que tiver registrado seu sistema, execute **up2date**, o agente de atualização do Red Hat, para controlar assinaturas, fazer o download de pacotes e alterar a configuração. Pode ser que você note a presença de um novo daemon, **rhnsd**, sendo executado em sua máquina. Trata-se do daemon que sonda o site do Red Hat em busca de novas atualizações.

Infelizmente, o Red Hat Network ainda não oferece todos os recursos que se queira de um serviço de atualização. Mais obviamente, não se pode usá-lo para atualizar para novos releases do Red Hat (por exemplo, fazer a atualização de 7.1 para 7.2). O equivalente do RHN no Debian, **apt-get**, já tem esse recurso há algum tempo e parece funcionar bem. **apt-get** é, em geral, um produto mais maduro do que o RHN, porém o RHN está evoluindo rapidamente.

23.6 APT-GET: DOWNLOAD E INSTALAÇÃO AUTOMÁTICOS

apt-get é um dos melhores recursos do Debian. Ele automatiza o download e a instalação de atualizações e de pacotes novos. Se um novo pacote depender de outros pacotes, **apt-get** os obterá e os instalará para você. Na realidade, com **apt-get** é possível atualizar um computador para a versão mais recente do Debian através de um único comando ou até mesmo fazer com que os seus computadores se auto-atualizem sem a intervenção de operadores. O sistema como um todo é conhecido como APT (Advanced Packaging Tool).

Já que ele é proveniente do universo Debian, o **apt-get** original suportava somente pacotes no formato **.deb**. Porém, agora ele foi portado para o mecanismo de pacotes RPM. Esta versão, conhecida como **apt-rpm**, se encontra disponível em

https://moin.conectiva.com.br/AptRpm

Infelizmente, **apt-rpm** é menos útil do que o **apt-get** original, pois sua infra-estrutura para o lado servidor não é tão bem desenvolvida. Para ter **apt-get** instalado e funcionando no Debian, basta apontar para o arquivo **/etc/apt/sources.list** no espelho Debian mais próximo (como padrão, ele aponta para a distribuição mestra). Esse nível de integração não está necessariamente presente em outras distribuições; provavelmente você terá de configurar o seu próprio repositório e fazer alguns ajustes para fazer com que **apt-get** funcione apropriadamente para as suas necessidades. Porém, se você quiser apenas implementar o seu próprio mecanismo de distribuição de pacotes local, **apt-get** e **apt-rpm** têm muito a oferecer.

A primeira regra para uso de **apt-get** em sistemas Debian (e, de fato, para todo o gerenciamento dos pacotes Debian) é ignorar a existência de **dselect**, que atua como um front-end para o sistema. Ele não é uma má idéia, mas a interface com o usuário é pobre.

Se estiver usando **apt-get** para controlar uma instalação básica do Debian por meio de um espelho-padrão Debian, a maneira mais fácil de ver quais pacotes estão disponíveis é visitar o site Web packages.debian.org. Esse site também fornece uma útil interface para pesquisa. Se você configurar o seu próprio servidor **apt-get**, então obviamente saberá quais pacotes estão disponíveis e poderá fornecer uma lista da maneira que quiser.

O Debian dispõe de vários pacotes vazios (a maioria com nomes começando por **task-**) que existem somente para listar outros pacotes como pré-requisitos. Já que **apt-get** faz o download automático e atualiza pacotes pré-requisitados conforme a necessidade, os pacotes **task-*** tornam mais fácil a instalação e a atualização de vários pacotes na forma de um só bloco. Por exemplo, instalar o pacote **task-gnome-desktop** garantirá que você terá todos os pacotes necessários para executar o GNOME.

Assim que tiver configurado o arquivo **sources.list** e souber o nome de um pacote desejado, a única coisa que fica faltando fazer é executar **apt-get update** para atualizar o cache de informações de pacotes de **apt-get**. Isso feito, basta executar **apt-get install** *nome_do_pacote* para instalar o pacote. O mesmo comando atualizará um pacote que já tenha sido instalado.

Snow666

Suponha que queiramos instalar uma nova versão do pacote **sudo** que corrija um bug de segurança. Primeiramente, sempre é mais inteligente fazer um **apt-get update**:

```
% sudo apt-get update
Get:1 http://http.us.debian.org potato/main Packages [824kB]
Get:2 http://non-us.debian.org potato/non-US/main Release [102B]
...
```

Agora podemos realmente recuperar o pacote. Note que estamos usando **sudo** à medida que recuperamos o novo pacote **sudo** — **apt-get** pode até atualizar pacotes que estão sendo usados no momento!

```
% sudo apt-get install sudo
Reading Package Lists... Done
Building Dependency Tree... Done
1 packages upgraded, 0 newly installed, 0 to remove and 191 not upgraded.
Need to get 0B/122kB of archives. After unpacking 131kB will be used.
(Reading database ... 24359 files and directories currently installed.)
Preparing to replace sudo 1.6.1-1 (using .../sudo_1.6.2p2-2_i386.deb) ...
Unpacking replacement sudo ...
Setting up sudo (1.6.2p2-2) ...
Installing new version of config file /etc/pam.d/sudo ...
```

Pronto! Nada mais fácil, não é mesmo?

Configurando apt-get

Configurar **apt-get** é objetivo e é uma área na qual a documentação Debian algumas vezes inconsistente está OK; praticamente tudo o que você precisa saber pode ser encontrado no *APT HOWTO:*

http://www.debian.org/doc/manuals/apt-howto

O arquivo de configuração **apt-get** mais importante é **/etc/apt/sources.list**, que informa **apt-get** onde obter seus pacotes. Cada linha especifica o seguinte: um tipo de pacote, atualmente `deb` ou `deb-src`; um URL que aponta para um arquivo, CD-ROM, servidor HTTP ou servidor FTP por meio do qual irá recuperar pacotes; uma "distribuição" que lhe permite prover várias versões dos pacotes (o Debian usa este campo para criar versões de suas distribuições, mas você poderá usá-lo para versões internas); e uma lista potencial de componentes — essencialmente, categorias de pacotes dentro de uma distribuição.

A menos que você queira configurar o seu próprio cache ou repositório APT, a configuração-padrão geralmente funciona bem. Se você tiver uma conexão de rede razoável, provavelmente irá querer colocar sinais de comentário nas linhas para os CD-ROMs Debian. Se quiser fazer o download de seu código-fonte, retire as os comentários das linhas que especificam `deb-src`. Enquanto estiver editando o arquivo, você deve alterar a identidade do espelho Debian para uma que esteja mais próxima a você; uma lista completa de espelhos pode ser encontrada em

http://www.debian.org/misc/README.mirrors

Para tornar as coisas incrivelmente fáceis, o Debian dispõe de uma ferramenta chamada **netselect-apt** que automaticamente irá gerar um arquivo **sources.list** para você; ele seleciona o espelho mais próximo que consegue achar, baseado no tempo de ping. **netselect-apt** faz parte do pacote **netselect**, que se encontra disponível no espelho Debian mais próximo.

Você também deve se certificar que security.debian.org esteja listado como uma fonte para que você tenha acesso às correções de segurança mais recentes.

Um exemplo de arquivo /etc/apt/sources.list

O exemplo a seguir usa http.us.debian.org para o arquivo estável, mas também adiciona non-us.debian.org como fonte para pacotes não americanos (materiais criptográficos que não podem ser exportados dos Estados Unidos). Acrescentamos security.debian.org, a fonte de todas as correções referentes à segurança e nosso servidor APT local, debserver. Finalmente, ativamos o download de código-fonte.

```
deb http://http.us.debian.org/debian stable main contrib non-free
deb http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free
deb http://security.debian.org stable/updates main contrib non-free
deb http://debserver/mypackages/ ./
deb-src http://http.us.debian.org/debian stable main contrib non-free
```

www.snow666.blogspot.com
Snow666


```
deb-src http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free
deb-src http://security.debian.org stable/updates main contrib non-free
```

Esse exemplo usa o repositório estável. Na prática, as distribuições Debian estáveis não são lançadas com tanta freqüência quanto as demais distribuições. Caso precise de pacotes das distribuições instáveis ou de teste, simplesmente copie as linhas `stable` e substitua `stable` por `unstable` (ou `testing`). As linhas são consultadas em ordem, portanto coloque as linhas `unstable` e `testing` no final do arquivo para dar prioridade às versões estáveis.

Usando proxies para ter escalabilidade em apt-get

Se você pretende usar **apt-get** num grande número de computadores, provavelmente irá querer colocar pacotes em cache locais — fazer o download de uma cópia de cada pacote para cada máquina não é um emprego sensato de largura de banda. Talvez você também precise direcionar seu **apt-get** através de um proxy caso o seu firewall requeira esta proteção adicional. Já que **apt-get** usa HTTP e FTP comuns, você pode usar qualquer proxy Web existente que por acaso possa ter instalado. **apt-get** honra a variável de ambiente http_proxy; da mesma forma, é possível configurar um proxy explícito por meio de uma linha no arquivo **/etc/apt/apt.conf**:

```
Acquire::http::Proxy "http://servidor.proxy.aqui:8080/
```

Uma alternativa a um proxy Web genérico é um pequeno aplicativo chamado **apt-proxy**. Apesar do nome, ele não é um verdadeiro proxy, mas sim um aplicativo que constrói um cache de pacotes aplicando **rsync** a eles por meio do verdadeiro servidor APT. **apt-proxy** está disponível em

http://sourceforge.net/projects/apt-proxy

Configurando um servidor APT interno

Em vez de usar um proxy, você também poderá configurar o seu próprio servidor APT autônomo e apontar seus clientes internos a ele. Esse modelo lhe permite ajustar os pacotes que você oferece a seus clientes, enviar facilmente atualizações (simplesmente instalar novas versões no servidor), distribuir seus próprios aplicativos na forma de pacotes e, mais importante ainda, fornecer suas próprias versões das distribuições.

Já que **apt-get** usa protocolos-padrão (HTTP e FTP) para fazer o download de seus pacotes, tudo o que você precisa para configurar um servidor APT é estabelecer um servidor Web ou FTP. Configurar um servidor NFS ou um CD-ROM itinerante para máquinas individuais também é viável; **apt-get** é muito flexível. Dado o amplo suporte para o emprego de firewalls e proxies http, provavelmente o http é o protocolo mais fácil de usar com o APT. Instruções para configuração do servidor Web Apache são apresentadas na seção 22.3.

Para transformar um servidor HTTP genérico num servidor APT, tudo o que é preciso é um diretório que contenha pacotes. Todo os pacotes podem estar num diretório ou ser organizados hierarquicamente como nos espelhos Debian.

Você também tem de gerar dois arquivos sintéticos de pacotes: **Packages.gz** e **Contents.gz**. **Packages.gz** é uma lista, em que se aplicou **gzip,** dos pacotes existentes no servidor e de suas dependências. **apt-get update** usa a lista para determinar o complemento de pacotes disponível. O arquivo **Contents.gz** associa arquivos brutos aos pacotes que os contêm; na verdade ele não é usado pelo próprio **apt-get**.

O comando **apt-ftparchive**, incluso no pacote **apt-utils**, gera automaticamente ambos os arquivos sintéticos. Assim que você os tiver criado, o restante é fácil. Uma linha como

```
deb http://debserver/mypackages/ ./
```

nos arquivos **/etc/apt/sources.list** dos clientes irá conectar **apt-get** ao seu servidor local. Execute **apt-get update** em cada cliente, depois use **apt-get** normalmente.

Para distribuir código-fonte, bem como pacotes binários, basta colocar os pacotes-fonte no servidor. Diferentemente de RPM, que possui um SRPM equivalente para pacotes-fonte, o Debian distribui os pacotes-fonte em três partes: o arquivo comum **.tar.gz**, um arquivo opcional **.diff.gz** (que é usado pelos mantenedores de pacotes para mostrar as alterações que eles fizeram em relação à distribuição original do código) e um arquivo **.dsc** contendo uma descrição do pacote. O código-fonte equivalente de **Packages.gz** é **Sources.gz**; ele também é gerado por **apt-ftparchive**.

O exemplo anterior não especifica um parâmetro de "distribuição". Se quiser usar nomes de distribuição próprios para permitir o emprego de versões internas, coloque cada versão num subdiretório e mude o `./` para o nome ou número da versão. A capacidade de emprego de versões é muito útil; é possível ter releases de "teste" e de "produção" diferentes que são simplesmente symlinks para as verdadeiras versões. Talvez você queira manter alguns

Snow666

computadores de teste que apontem para a sua distribuição-teste. Quando estiver satisfeito com a versão, atualize suas máquinas de produção apontando o symlink de produção para a nova versão.

Automatizando apt-get

É possível executar **apt-get** segundo um cronograma regular por meio de **cron**. Mesmo que você não instale pacotes automaticamente, talvez você queira executar **apt-get update** regularmente para manter os sumários de pacotes atualizados.

apt-get dist-upgrade fará o download e instalará novas versões de quaisquer pacotes que sejam instalados na máquina local. **dist-upgrade** é similar a **upgrade** mas tem uma manipulação de dependências ligeiramente mais inteligente. Se quiser brincar com fogo, faça com que máquinas executem essa atualização automaticamente — use a opção **-yes** para operação sem a presença de operador; ela responde qualquer pergunta de confirmação que o **apt-get** possa fazer com um entusiástico "Sim!".

Provavelmente talvez não seja uma boa idéia realizar atualizações automatizadas diretamente dos espelhos Debian. Porém, caso tenha servidores APT, pacotes e sistemas de controle de versão próprios, essa é uma maneira perfeita para manter clientes sincronizados. Um pequeno script de shell como o mostrado a seguir manterá um computador atualizado com seu servidor APT:

```
#!/bin/sh
apt-get update
apt-get -yes dist-upgrade
```

Chame esse script por meio de um serviço **cron** caso queira rodá-lo a cada noite. Você também pode se referir a ele por meio de um script de inicialização de sistema para fazer com que a máquina atualize no momento da inicialização. Veja o Capítulo 9, *Processos Periódicos*, para obter maiores informações sobre **cron**; veja o Capítulo 2, *Inicialização e Desligamento*, para obter maiores informações sobre scripts de inicialização.

Se você executar atualizações a partir de **cron** em diversas máquinas, você deverá usar randomização de tempo para certificar-se de que todo mundo não tente atualizar de uma só vez. O pequeno script em Perl na página 496 pode ajudá-lo nesta tarefa.

Se você não confia muito no fonte dos pacotes, considere fazer o download automático de todos os pacotes alterados sem instalá-los. Use a opção **--download-only** de **apt-get** para solicitar esse comportamento, depois reveja os pacotes manualmente e instale aqueles que deseja atualizar. Os pacotes obtidos via download são colocados em **/var/cache/apt** e ao longo do tempo este diretório pode acabar ficando muito grande. Elimine os arquivos que não são usados deste diretório através de **apt-get autoclean**.

23.7 Leitura Recomendada

BAILEY, ED. *Maximum RPM*. Indianapolis: SAMS. 1997.
BURGESS, MARK. "Cfengine: A Site Configuration Engine." USENIX Computing Systems, v. 8, n. 3. 1995.
INTEL CORPORATION AND SYSTEMSOFT. Preboot Execution Environment (PXE) Specification. ftp://download.intel.com/ial/wfm/pxespec.pdf.
MINTHA, JIM; PIETER KRUL. "UltraLinux FAQ." www.ultralinux.org/faq.html.
OETIKER, TOBIAS. "SEPP: Software Installation and Sharing System." LISA 1998 proceedings.
"PXELinux Questions." syslinux.hackerdojo.com/pxe.php.
RODIN, JOSIP. *Debian New Maintainers' Guide*. www.debian.org/doc/maint-guide.
Este documento contém excelentes informações sobre pacotes **.deb**.
SILVA, GUSTAVO NORONHA. "APT HOWTO." www.debian.org/doc/manuals/apt-howto.
STÜCKELBERG, MARC VUILLEUMIER; DAVID, CLERC. "Linux Remote-Boot mini-HOWTO: Configuring Remote-Boot Workstations with Linux, DOS, Windows 95/98 and Windows NT." www.linux-doc.org/HOWTO/mini/Remote-Boot.html.
WACHSMANN, ALF. "How to Install Red Hat Linux via PXE and Kickstart." www.stanford.edu/~alfw/PXE-Kickstart/PXE-Kickstart.html.

23.8 Exercícios

E23.1 Aponte as diferenças entre Kickstart, YaST e SystemImager. Quando você escolheria SystemImager em relação a um dos instaladores específicos de uma dada distribuição?

Snow666

E23.2 Suponha que você e um outro membro de sua esquadra de elite estivessem usando RCS para controlar um projeto de software. Seu parceiro se esqueceu de registrar as modificações de um arquivo crítico que agora você precisa modificar. Como você poderia desfazer o bloqueio e não perder nenhuma alteração feita pelo seu companheiro?

★ **E23.3** Descreva as etapas necessárias para criar um pacote RPM. Use este procedimento para empacotar um produto de software de uma de suas outras classes de computação.

★ **E23.4** Repita o exercício E23.3, porém crie um pacote de formato Debian **.deb**.

E23.5 Reveja a maneira pela qual o software local é organizado em sua instalação. O sistema poderá ser ampliado? Ele é fácil de ser usado? Discuta.

★★ **E23.6** Configure o instalador de rede de sua preferência e instale uma máquina nova usando o seu servidor. Descreva todas as etapas necessárias para realizar esta tarefa. Quais são alguns dos empecilhos encontrados? Quais são alguns dos problemas de escalabilidade que você descobriu com o instalador escolhido?

www.snow666.blogspot.com
Snow666

24 Impressão

Imprimir normalmente é uma das funções mais irritantes e difíceis do Linux a serem suportadas por um administrador de sistemas. Os usuários muitas vezes tomam por certo a impressão, mas emitir páginas perfeitamente apresentadas numa impressora que se encontra a três metros do usuário pode exigir uma série de contorcionismos administrativos desafiadores.

Anos atrás, as impressoras mais comuns eram impressoras de linha ASCII. As impressoras laser eram novas, caras e raras. Dispositivos de saída de alta resolução exigiam drivers e programas de formatação personalizados.

No final do milênio, as impressoras de linha tinham praticamente se tornado peças de museu. Foram estabelecidos numerosos padrões para linguagens de impressão e descrição de páginas. As impressoras laser permearam o mercado e eram largamente usadas.

Hoje, as impressoras laser normalmente se conectam a uma rede Ethernet em vez de uma porta serial ou paralela. Elas perderam de longe o mercado low-end para as impressoras jato de tinta.

Com todas essas mudanças na tecnologia, era de esperar que os sistemas de impressão Linux fossem extremamente flexíveis e, de fato, são. Infelizmente, os sistemas mais populares ainda se baseiam no antigo software usado para as impressoras de linha de antigamente. Eles foram mexidos e sobrecarregados numa tentativa de suportarem as novas tecnologias e o resultado normalmente é de pouca qualidade.

Neste capítulo começamos com discussões genéricas sobre a terminologia do mundo da impressão e os sistemas de impressão Linux, depois apresentamos instruções para configuração de impressoras em cada um de nossos sistemas de referência. Finalmente, concluímos com um breve guia para a depuração de impressão, uma discussão dos softwares mais comuns para impressão e algumas dicas gerais sobre impressão.

24.1 Miniglossário de Termos de Impressão

Embora uma visão geral sobre a tecnologia atual para impressão esteja fora do escopo deste livro, tentaremos dar-lhe informações suficientes para responder quando alguém começa, ao falar, utilizar jargão de impressão.

spooler Um spooler é um software que recebe tarefas de impressão, as armazena, as prioriza e as envia seqüencialmente a uma impressora. Um comando em nível de usuário submete tarefas ao spooler para impressão. Um spooler é normalmente chamado de servidor de impressão. Algumas impressoras possuem seus próprios spoolers internos.

PDL A maioria das impressoras aceita entrada em uma ou mais "linguagens de descrição de página" que especificam as imagens a serem colocadas na página de uma forma abstrata. Descrições PDL são mais eficientes para serem transmitidas do que imagens brutas, e elas são mais fáceis para os apli-

Snow666

cativos gerarem. Elas também têm o benefício de serem independentes de dispositivos e de resolução. As PDLs mais conhecidas são PostScript, PCL e PDF.

bitmap Algumas vezes precisamos imprimir imagens que não são facilmente descritas numa PDL. Nesses casos, você usa um bitmap, que é um conjunto de dados que especifica quais pontos são preenchidos e quais não são (ou em que cor cada ponto se encontra, no caso de uma imagem colorida ou em tons de cinza).

Como acontece com as PDLs, há vários formatos concorrentes para armazenamento de bitmaps. Cada PDL suporta pelo menos um formato. Já que os bitmaps normalmente são muito grandes, eles normalmente são compactados. Entre os formatos bitmap mais comuns temos: JPEG, PNG, TIFF e GIF.

RIP Um RIP (Raster Image Processor, processador de imagens rasterizadas) é um sistema de software que aceita documentos em uma ou mais PDL e os converte para um formato bitmap apropriado para determinado dispositivo de saída. A conversão PDL-bitmap é normalmente feita por um RIP dentro da impressora.

filtros Filtros são programas que modificam as tarefas de impressão na rota do spooler para a impressora. Os filtros convertem formatos de arquivo, mantêm registros contábeis e muitas vezes tratam dos processos de comunicação com a impressora. Os filtros normalmente não são necessários com impressoras de texto simples, mas são essenciais para enviar tarefas para as impressoras que requerem PDLs não padronizadas. Algumas impressoras PostScript são capazes de lidar com entrada não filtrada, porém a maioria prefere tarefas de impressão especialmente filtradas. Veja a página 421 para obter maiores informações sobre filtros.

PostScript PostScript é de longe a PDL mais comum encontrada em sistemas Linux. Ela foi originalmente desenvolvida pela Adobe Systems e a maioria das impressoras PostScript usa um interpretador licenciado pela Adobe.

Quase todos os programas de layout de páginas são capazes de gerar PostScript.

O PostScript é, na verdade, uma linguagem de programação madura. Podemos ler programas PostScript com um editor de texto ou com **more**. Os programas contêm um número enorme de parênteses, chaves e barras e normalmente começam com os caracteres `%!PS`. Embora esses caracteres de inicialização não sejam exigidos pela linguagem em si, os programas Linux para impressão normalmente os procuram ao tentarem classificar tarefas de impressão.

PCL Printer Command Language é uma alternativa da HP ao PostScript. Ela é encontrada quase que exclusivamente em impressoras HP e é bem comum no mundo dos PCs. Aplicativos Linux normalmente não são capazes de gerar diretamente PCL, de modo que eles requerem um filtro para conversão.

PDF O Portable Document Format da Adobe é comumente produzido e usado pelo Adobe Acrobat. Trata-se de um formato independente de plataforma que é normalmente usado para troca eletrônica de documentos tanto para uso on-line quanto para uso off-line (impresso). Um aplicativo gratuito para leitura deste formato (para o Linux e tantos outros sistemas operacionais) está disponível em www.adobe.com.

24.2 Sistemas de Impressão Linux

Diversos sistemas de impressão diferentes se encontram disponíveis para uso sob o Linux. Neste capítulo revisamos as opções mais comuns. Para informações adicionais, veja o site Web www.linuxprinting.org, que mantém um vasto conjunto de recursos para impressão no Linux.

O sistema de impressão LPD (Line Printer Daemon) se baseia no sistema de impressão Berkeley (BSD), que tem sido um esteio do ambiente UNIX há anos. A maioria das demais versões do UNIX ainda usa alguma forma de sistema LPD hoje em dia. O LPD é o sistema de impressão no SuSE.

LPRng é uma versão de próxima geração do LPD com flexibilidade e funcionalidade adicional desenhada para o mundo moderno da impressão. Esse sistema é o padrão para os sistemas Red Hat e Debian.

LPD e LPRng são extremamente similares em termos de abordagem e de configuração e eles são certamente os sistemas de impressão mais comumente usados em ambientes profissionais. Cobrimos estes sistemas de forma detalhada mais à frente, ainda neste capítulo.

www.snow666.blogspot.com
Snow666

Dois outros sistemas de impressão disponíveis para o Linux que vale a pena mencionar são o PDQ e o CUPS. O PDQ, encontrado em pdq.sourceforge.net, adota uma abordagem exclusiva e inovadora de controle descentralizado sobre as impressoras (a abordagem LPD/LPRng é por natureza mais voltada ao ambiente burocrático governamental). O PDQ é capaz de fornecer a flexibilidade tão necessária em ambientes dinâmicos e se diz que é mais simples e mais confiável do que sistemas tradicionais.

Um outro sistema promissor, CUPS (Common UNIX Printing System), foi portado para o Linux e pode ser adequado para ambientes que suportam impressão entre plataformas numa ampla gama de variantes UNIX. Informações sobre o CUPS podem ser encontradas em www.cups.org.

24.3 Tipos de Impressoras

O Linux permite que coloquemos em spool tarefas para quase todo tipo de impressora. No nível mais básico, as impressoras são classificadas conforme suas interfaces de conexão (rede, seriais, paralelas) e pelo tipo de dados que elas compreendem (texto, PostScript, PCL ou algo inteiramente diferente).

Muitas das impressoras "baratas" usadas nos sistemas Windows (conhecidas coletivamente como WinPrinters) não podem ser usadas no Linux. Essas impressoras possuem pouca inteligência embutida e não são capazes de entender qualquer PDL. Algumas das informações necessárias para se comunicar com essas impressoras foram ocultas em código de driver proprietário. Tais segredos obstruem tentativas de se desenvolver suporte Linux para esses dispositivos, embora a comunidade Linux tenha demonstrado uma capacidade admirável em termos de engenharia reversa. Muitas WinPrinters *são* bem suportadas no Linux, mas verifique primeiro e compre depois.

Impressoras seriais e paralelas

As impressoras seriais requerem alguma configuração extra. O software spooler precisa saber os valores apropriados para taxa de transmissão e outras opções seriais para poder se comunicar apropriadamente com a impressora. Veja a seção 24.4 para obter maiores informações sobre como especificar estes detalhes, e o Capítulo 7, *Dispositivos Seriais*, para uma visão mais abrangente.

As portas paralelas são mais rápidas do que as portas seriais padrão e, felizmente, para os administradores de sistemas, elas requerem a configuração de um número menor de opções. Embora o padrão não tenha amadurecido graciosamente, ele efetivamente nos fornece portas que requerem relativamente poucos ajustes. No Linux, a primeira (e normalmente a única) porta paralela é **/dev/lp0**.

Uma tecnologia serial melhor e mais rápida chamada USB (Universal Serial Bus, barramento serial universal) também abriu seu caminho no mundo Linux. O USB se tornou incrivelmente popular no Windows, mas o suporte ao Linux ainda é um tanto inconsistente. Verifique o banco de dados de dispositivos USB suportados em www.qbik.ch/usb/devices ou www.linux-usb.org para ver a situação de seu hardware.

Impressoras de rede

Algumas impressoras contêm interfaces de rede totalmente maduras, que permite que elas se conectem diretamente numa rede e aceite tarefas através de uma ou mais redes ou protocolos de impressão. Os dados podem ser enviados às impressoras conectadas à rede de maneira muito mais rápida do que se fossem enviados a impressoras em portas seriais ou paralelas.

Pelo fato de qualquer computador da rede poder potencialmente gerar um spool diretamente para a impressora de rede, surgem problemas de disputas. O controle administrativo em geral está faltando.

Para simplificar a administração, você deve tentar configurar a rede de forma que apenas alguns hosts controlem todas as impressoras. Outras máquinas devem simplesmente transmitir tarefas para essas máquinas servidoras de impressão. Esse tipo de configuração pode lhe poupar trabalho, pois você não terá de ficar de olho no sistema de impressão em todas as máquinas. Além disso, você terá relativamente menos configurações para analisar quando ocorrer algum problema de impressão.

Muitas impressoras de rede a laser incluem um servidor LPD que roda dentro da impressora. Esse recurso permite que clientes Linux coloquem arquivos em spool para a impressora exatamente da mesma maneira que eles gerariam arquivos spool para outro servidor Linux. Já que praticamente todas as variações do UNIX e do Linux são capazes de gerar spool para um servidor LPD, gostamos muito desse tipo de impressora.

Impressoras de rede mais antigas exigiam que as tarefas de impressão fossem enviadas para a porta TCP 9100. Essa configuração é difícil de ser suportada com o sistema de impressão LPD (que vem com o SuSE), mas é fácil com o LPRng (que vem com o Red Hat e o Debian).

Uma outra opção para impressão em rede é usar o CIFS (Common Internet File System), que normalmente é conhecido como Samba em homenagem à sua implementação UNIX e Linux mais popular. O CIFS permite que as

Snow666

impressoras sejam compartilhadas por meio de servidores UNIX, Linux e Windows, de forma que qualquer um destes computadores poderá atuar como um servidor de impressão de rede.[1] O CIFS é especialmente útil quando precisamos compartilhar impressoras UNIX com desktops Windows. Veja o Capítulo 27, *Cooperando com o Windows*, para obter mais informações a respeito do CIFS.

A vida sem o PostScript

As impressoras PostScript são bem suportadas pelos sistemas de impressão Linux e a configuração destas impressoras é relativamente fácil. Infelizmente, impressoras não-PostScript como as jato de tinta e algumas impressoras laser baratas são ligeiramente mais difíceis de lidar.

*Consulte a seção 24.8 para obter maiores informações sobre **ghostscript**.*

Para imprimirmos numa impressora não-PostScript, normalmente precisamos de software especial para converter a tarefa de impressão na PDL preferida da impressora. Alguns fabricantes podem fornecer o software Linux apropriado; esta é, geralmente, a melhor opção caso ela esteja disponível. Uma alternativa popular é o pacote **ghostscript**, capaz de converter PostScript para as PDLs proprietárias de centenas de impressoras. Você precisará especificar um filtro de impressão para chamar adequadamente **ghostscript** durante a execução.

24.4 LPD: O Velho e Bom Sistema de Impressão

O sistema de impressão LPD foi desenvolvido especificamente para emprego com impressoras de linha. Felizmente, seu desenho permitiu que o sistema fosse expandido para suportar a maioria das PDLs e impressoras atuais. A parte referente à rede do sistema de impressão LPD também se estende bem para redes grandes e heterogêneas e permite que vários computadores compartilhem impressoras. O spooler de impressão se tornou de tal maneira um padrão que ele trilhou o seu caminho dentro de várias impressoras de rede.[2]

Uma visão geral do processo de impressão

O acesso a impressoras é controlado pelo daemon **lpd**, que reside em **/usr/sbin** e é normalmente iniciado no momento da inicialização. **lpd** é responsável por aceitar tarefas de impressão de usuários e outros **lpd**s (remotos), por processar tais tarefas e encaminhá-las a uma impressora de fato. Para conseguir realizar essas tarefas, **lpd** lê as informações de configuração da impressora de **/etc/printcap**, o banco de dados de informações de impressão do sistema.

Os usuários chamam o programa **lpr** para submeterem suas tarefas de impressão a **lpd**. Esses dois processos comunicam-se através do "pipe com nome" (named pipe) **/dev/printer**.

Ao determinar para qual impressora enviar a tarefa, **lpr** analisa primeiramente a linha de comando. Se o argumento **-P***impressora* for passado a **lpr**, *impressora* se torna o destino. Caso contrário, **lpr** verifica o ambiente para ver se a variável PRINTER está definida e, em caso positivo, **lpr** usa o valor desta variável. Se tudo isso falhar, a tarefa será submetida à impressora-padrão do sistema, que é a impressora chamada lp, ou se não houver nenhuma lp, à primeira entrada contida no arquivo **/etc/printcap**. Quase todos os comandos relacionados com impressão, inclusive **lpq** e **lprm**, entendem a variável de ambiente PRINTER e o argumento **-P**.

Tão logo **lpr** souber para onde a tarefa atual é direcionada, ele procura a impressora no arquivo **/etc/printcap**. Esse arquivo designa o diretório onde **lpr** deve colocar as tarefas de impressão para aquela impressora. Esse "diretório de spool" normalmente é **/var/spool/lpd/***nome_da_impressora*.

lpr cria dois arquivos no diretório de spool para cada tarefa. O nome do primeiro arquivo é formado pelas letras **cf** (control file, arquivo de controle) seguidas por um número que identifica a tarefa.[3] Esse arquivo contém informações de referencia e de manipulação para a tarefa, como a identidade do usuário que a submeteu. A parte numérica do nome do arquivo tem espaço para somente três dígitos, de maneira que o sistema de impressão acaba ficando confuso se mais de 999 tarefas forem colocadas na fila. O nome do segundo arquivo começa com **df** (data file, arquivo de dados) seguido pelo mesmo número. Esse arquivo contém os dados reais a serem impressos. Depois de o arquivo ter sido colocado em spool, **lpr** notifica o daemon **lpd** da existência da tarefa.

---

1. Entretanto, não recomendamos essa configuração para impressoras com volumes extremamente altos devido ao risco de bloqueios de arquivos.
2. O padrão LPD foi documentado (após o fato) na RFC1179. Um padrão mais novo, o IPP (Internet Printing Protoocol), é detalhado na RFC2568 e coligadas. Ele é similar, mas ainda não foi adotado em larga escala. O sistema de impressão CUPS usa IPP.
3. O arquivo **cf** é, na verdade, chamado **tf** ("temporary file", "arquivo temporário") enquanto **lpr** está no processo de aceitar uma tarefa. Depois de o arquivo ter sido gravado, **lpr** altera o nome do arquivo de **tf***xxx* para **cf***xxx*.

Snow666

Quando **lpd** recebe essa notificação, ele consulta o arquivo **printcap** para determinar se o destino é local ou remoto. Se a impressora estiver conectada localmente, **lpd** faz uma verificação para ter certeza de que um daemon de impressão está sendo executado na fila da impressora apropriada e cria uma (criando uma cópia de si mesmo) se necessário.

Se a impressora solicitada estiver conectada a uma máquina diferente, **lpd** abre uma conexão para o **lpd** da máquina remota e transfere tanto o arquivo de controle como os dados. **lpd** elimina então as cópias locais desses arquivos.

O escalonamento de tarefas de impressão é feito segundo um esquema "o primeiro que entra é o primeiro que sai", porém o administrador do sistema pode modificar o cronograma de impressão usando **lpc** nas tarefas desejadas. Infelizmente, não há nenhuma forma de instruir permanentemente o sistema de impressão para dar tratamento preferencial a tarefas em spool de um determinado usuário ou máquina.

Quando a tarefa já está pronta para ser impressa, **lpd** cria uma série de canais Linux entre o arquivo spool e o hardware de impressão que transporta os dados a serem impressos. No meio desse canal **lpd** instala um processo de filtragem que pode revisar e editar o conteúdo do fluxo dos dados antes de atingir a impressora.

Os processos de filtragem podem realizar várias transformações nos dados ou não fazer nada neles. Seus objetivos básicos são o de fornecer formatação e o de suportar quaisquer protocolos específicos a determinados dispositivos que podem ser necessários para lidar com uma dada impressora. O filtro-padrão de uma impressora é especificado em **/etc/printcap**, mas o filtro-padrão pode ser sobreposto na linha de comando **lpr**.

Controlando o ambiente de impressão

Para manutenção diária do sistema de impressão, você precisa somente de três comandos: **lpq**, **lprm** e **lpc**. **lpq** examina a fila das tarefas que estão aguardando para serem impressas numa determinada impressora. **lprm** elimina uma ou mais dessas tarefas, apagando seus arquivos de dados armazenados e removendo quaisquer referências a eles de dentro do sistema de impressão. Ambos os comandos estão disponíveis a usuários e ambos funcionam através de uma rede (se você tiver sorte).

lpc permite que você faça uma série de alterações no ambiente de impressão como desabilitar impressoras e reordenar filas de impressão. Embora algumas de suas funções estejam disponíveis para usuários, **lpc** é, basicamente, uma ferramenta administrativa. A Tabela 24.1 mostra alguns outros comandos e daemons associados ao sistema de impressão LPD.

Tabela 24.1 Comandos de impressão de LPD.

Comando	Localização	Função
lpq	**/usr/bin**	Mostra o conteúdo e o estado atual das filas de impressão
lpr	**/usr/bin**	Coloca tarefas em fila para impressão
lprm	**/usr/bin**	Cancela uma tarefa de impressão ou colocada em fila
lpc	**/usr/sbin**	Controla uma impressora ou fila
lpd	**/usr/sbin**	Agenda e imprime tarefas
lptest	**/usr/bin**	Gera um padrão de teste ASCII
lpunlock	**/usr/bin**	Desbloqueia impressoras emperradas

lpd: o spooler de impressão

Quando **lpd** é iniciado pela primeira vez, ele lê o arquivo **/etc/printcap**, no qual as impressoras de sistema estão definidas. A seguir ele começa a imprimir quaisquer tarefas que estão aguardando no diretório de spool e começa a escutar para ver se há novas solicitações de impressão.

Se você iniciar **lpd** com o flag **-l**, ele registrará em log solicitações de impressão através de syslog sob o recurso "lpr". Sem o flag **-l**, **lpd** registra em log somente erros do sistema de impressão.

*Consulte a seção 21.7, Capítulo 21, para obter maiores informações sobre o arquivo **hosts.equiv**.*

O controle de acesso é definido para cada host; o sistema de impressão LPD não é capaz de suportar controle de acesso para usuários remotos específicos.[4] Somente hosts cujos nomes aparecem nos arquivos **/etc/hosts.equiv** ou **/etc/hosts.lpd** têm permissão para colocar tarefas de impressão em spool.

4. Na verdade, os filtros de impressão são capazes de realizar esse tipo de autenticação. Mas pelo fato de a maioria dos sistemas usar uma série de filtros, manter um controle de acesso uniforme entre eles se torna impraticável.

Snow666

Lembre-se de que acrescentar um nome de host a **/etc/hosts.equiv** indica total confiança nesse host. Recomendamos que você adira ao arquivo **/etc/hosts.lpd** para controle de acesso a impressoras. O LPRng dispõe de um modelo de segurança mais refinado.

lpr: submeter tarefas de impressão

lpr é o único programa num sistema LPD capaz de colocar informações em filas para impressão. Outros programas que provocam a impressão de arquivos (por exemplo, **enscript** e Acrobat) têm de fazer isto chamando **lpr**.

Podem ser especificadas várias opções úteis como argumentos de **lpr**. O flag **-#***num* produz *num* cópias, e o flag **-h** suprime a página de cabeçalho. Remanescente da era das impressoras lentas, o flag **-m** solicita que seja enviado um e-mail para o proprietário quando a tarefa de impressão tiver sido completada.

Por exemplo, para imprimir duas cópias de um arquivo chamado **thesis** numa impressora chamada **howler-lw**, você digitaria

```
% lpr -Phowler-lw -#2 thesis
```

lpq: visualizar a fila de impressão

lpq é usado normalmente apenas com a opção **-P** para selecionar uma impressora, embora o flag **-l** esteja disponível para gerar saída mais detalhada. A saída de **lpq** assemelha-se ao seguinte:

```
% lpq
anchor-lj is ready and printing
Rank     Owner    Job   Files           Total Size
active   garth    314   domain.2x1.ps   298778 bytes
1st      kingery  286   standard input  17691 bytes
2nd      evi      12    appendices      828 bytes
3rd      garth    13    proc            43229 bytes
4th      scott    14    periodic        16676 bytes
5th      garth    16    standard input  489 bytes
```

A primeira coluna lhe informa a ordem na qual as tarefas serão impressas. Essas informações são bem supérfluas, pois as linhas de saída sempre estão em ordem, com a tarefa ativa no topo da fila e a última delas na parte inferior. Se a primeira tarefa estiver listada como `1st` em vez de `active`, não está sendo executado nenhum daemon de impressão na fila.

A segunda coluna lhe informa qual usuário gerou spool para cada tarefa. A terceira coluna dá o número de identificação da tarefa para cada uma delas; é importante conhecer esse número caso pretenda manipular a tarefa mais tarde com **lprm** ou **lpc**. A quarta coluna mostra os nomes de arquivos que foram listados na linha de comando **lpr** usado para jogar a tarefa em spool. Se os dados forem provenientes de um canal (como acontece com a primeira e a quinta tarefas acima), a entrada nessa coluna será `standard input`. A quinta e última coluna informa o tamanho da tarefa. Esse número indica o tamanho da tarefa antes de ser enviada para o programa de filtragem e não dá nenhuma informação sobre o número de páginas que uma tarefa terá ou quanto tempo ela levará para ser impressa.

lprm: eliminar tarefas de impressão

A forma mais comum de **lprm** é **lprm** *jobid*, onde *jobid* é a identificação do número de tarefa relatado por **lpq**. **lprm** *usuário* elimina todas as tarefas pertencentes a *usuário*. **lprm** sem o emprego de argumentos elimina a tarefa ativa. **lprm –** elimina todas as tarefas submetidas; se você for root, ele elimina todas as tarefas presentes na fila. Nenhum usuário comum pode eliminar as tarefas de um outro usuário, mas o superusuário pode eliminar qualquer tarefa.

O sistema de impressão preserva uma noção da origem de uma tarefa, bem como do usuário que o colocou em spool, e o processo correspondente do **lprm** leva estas duas informações em consideração. Portanto, garth@sigi não é equivalente a garth@boulder, e nenhum deles pode eliminar as tarefas do outro.

Tentar aplicar **lprm** à tarefa ativa pode gerar problemas em algumas impressoras. O processo de filtragem para a tarefa não é notificado apropriadamente do término, fazendo com que todo o sistema entre num estado de parada, com o processo de filtragem detendo um bloqueio exclusivo sobre a porta da impressora e impedindo que outros processos usem a impressora.

A única maneira de corrigir essa situação é usar **ps** para identificar os processos de filtragem e eliminá-los manualmente. **lpc** não é de nenhuma valia nessa situação. Inicializar o sistema sempre curará uma situação de impressora pendurada, porém se trata de uma medida drástica. Antes de apelar para uma reinicialização, elimine e reinicie a cópia mestra de **lpd** e, manualmente, remova tarefas do diretório de spool usando o comando **rm**.

www.snow666.blogspot.com
Snow666

lpc: fazer alterações administrativas

O comando **lpc** é capaz de realizar as seguintes funções:

- Habilitar ou desabilitar filas para determinada impressora
- Habilitar ou desabilitar a impressão em determinada impressora
- Eliminar todas as tarefas da fila de uma impressora
- Mover uma tarefa para o topo da fila de uma impressora
- Iniciar, parar ou reiniciar o daemon **lpd**
- Obter informações sobre o estado da impressora

Quando o sistema de impressão está operando bem, **lpc** funciona perfeitamente. Mas assim que um filtro emperrar ou surgir algum outro pequeno problema, **lpc** fica totalmente baratinado. E ele mente: algumas vezes ele diz ter resolvido tudo quando, na realidade, não fez nada. Talvez você tenha de corrigir as coisas manualmente ou até mesmo desligar e ligar o equipamento quando se chega nesta situação.

lpc não pode ser usado através da rede; você tem de logar na máquina que está de posse da impressora que você quer manipular. **lpc** normalmente é usado interativamente, embora também se possa chamá-lo num modo imediato colocando um dos comandos interativos na linha de comando de **lpc**. Assim que tiver ativado lpc, os vários comandos descritos a seguir estarão disponíveis:

help [*comando*]

help sem argumentos exibe uma lista de todos os comandos **lpc** disponíveis. Com um argumento, ele exibe uma descrição de uma linha de determinado comando.

enable *impressora*
disable *impressora*

Esses comandos habilitam ou desabilitam o spooling de tarefas para a impressora citada. Os usuários que tentarem colocar arquivos em fila serão cortesmente informados de que o spooling foi desabilitado. As tarefas que já estiverem na fila não serão afetadas. Você realiza essa operação simplesmente ativando ou desativando a permissão de execução de grupo em **/var/spool/lpd/***impressora***/lock**.

start *impressora*
stop *impressora*

start habilita e **stop** desabilita a impressão da impressora citada. As tarefas de impressão ainda podem ser colocadas em spool quando uma impressora tiver sido parada, mas elas não serão mais impressas até que a impressão seja reiniciada. **start** e **stop** operam ativando ou desativando a permissão de execução do proprietário sobre **/var/spool/lpd/***impressora***/lock**. Eles também iniciam e eliminam os daemons apropriados para a impressora. **stop** permite que a tarefa ativa seja completada antes de se desabilitar a impressão.

abort *impressora*

abort é exatamente igual à **stop**, mas ele não permite que a tarefa ativa seja completada. Quando a impressão for reabilitada, esta tarefa será impressa novamente.

down *impressora mensagem*
up *impressora*

Esses comandos afetam tanto o spooling quanto a impressão. Use-os quando uma impressora estiver realmente com defeito ou tiver de ser retirada de operação por um longo período. O parâmetro *mensagem* fornecido para **down** pode ter o comprimento que você quiser (em uma linha) e precisa ser colocado entre aspas; ele será posto no arquivo **/var/spool/lpd/***impressora***/status** da impressora e mostrado aos usuários quem executou **lpq**. Normalmente você vai querer usar esse recurso para registrar uma breve explanação do porquê a impressora está indisponível e quando ela voltará a funcionar. O comando **up** reverte o efeito de um comando **down**.

clean *impressora*

Esse comando elimina todas as tarefas da fila de impressão, inclusive a tarefa atual. Pelo fato de o daemon de impressão para a fila ainda armazenar referências para os arquivos da tarefa atual, o Linux não irá realmente eliminá-los, e a tarefa atual será completada.

Snow666

topq *impressora jobid*
topq *impressora nome-do-usuário*

A primeira forma desloca a tarefa especificada para o topo da fila de impressão. A segunda forma promove todas as tarefas pertencentes a *nome-do-usuário*.

restart *impressora*

O comando **restart** reinicializa o daemon de impressão que morreu misteriosamente. Você saberá que o daemon está morto quando **lpq** o informar "no daemon present". Embora você possa pensar que **restart** terá o mesmo efeito do que um **stop** seguido de um **start**, isto não acontecerá; **restart** falhará em reiniciar uma impressora que ainda tem um filtro em execução.

status *impressora*

O comando **status** mostra quatro coisas referentes a uma impressora: se o spooling está habilitado ou não, se a impressão está habilitada ou não, o nuúmero de entradas na fila e o estado do daemon para aquela impressora. Se não houver nenhuma entrada na fila, você verá algo como o seguinte.

```
lpc> status cer
cer:
queuing is enabled
printing is enabled
no entries
no daemon present
```

O fato de nenhum daemon estar presente não é motivo para preocupação; daemons específicos a determinadas impressoras se retiram após a fila estar vazia e não são reinicializados pela cópia mestra de **lpd** até uma outra tarefa ser colocada em spool.

O arquivo /etc/printcap

/etc/printcap é o banco de dados mestre do sistema de impressão LPD. Ele contém as informações necessárias para imprimir em impressoras locais e remotas. Uma impressora tem de ser descrita no arquivo **printcap** antes das tarefas poderem ser submetidas a ele.

O primeiro item em cada entrada **printcap** é uma lista de nomes para a impressora, separados por barras verticais. Os nomes são seguidos por uma série de ajustes de configuração separados por dois-pontos. As opções de configuração são no formato xx, xx=*string* ou xx#*número*, onde xx é o nome de dois caracteres de um parâmetro e *string* e *número* são valores a serem atribuídos a ele. Quando não for atribuído nenhum valor, a variável é booleana e sua presença indica "verdadeiro".

A instrução null é aceitável, de modo que você pode colocar dois-pontos lado a lado. Ela é útil para iniciar e terminar cada linha com um sinal de dois pontos, para tornar mais fácil modificações subseqüentes. Comentários em **/etc/printcap** iniciam com um sinal #. As entradas podem ocupar várias linhas se as linhas intermediárias forem terminadas com uma barra invertida. As linhas de continuação são, por convenção, recuadas.

A sintaxe do arquivo **printcap** é ilustrada no exemplo a seguir, que define uma impressora remota conectada à máquina anchor:

```
# HP LaserJet 5M remote printcap. CS Department.

anchor-lj|cer|1-56|LaserJet 5M in cer lab:\
   :lp=/var/spool/lpd/anchor-lj/.null:\
   :sd=/var/spool/lpd/anchor-lj:\
   :lf=/var/adm/lpd-errs:\
   :rw:mx#0:rm=anchor:rp=anchor-lj:
```

Da primeira linha, podemos observar que "cer", "anchor-lj", "1-56" e "LaserJet 5M in cer lab" são todos nomes equivalentes para a mesma impressora. Esses nomes são o nome da impressora, uma abreviatura bem conhecida, o número da sala onde se encontra a impressora e uma descrição completa. Embora você possa atribuir às suas impressoras quantos nomes quiser, inclua pelo menos três formas do nome primário:

- Nome abreviado – três ou quatro caracteres, fáceis de serem digitados (por exemplo, "cer")
- Nome completo – nome de host e tipo de impressora (por exemplo, "anchor-lj")
- Nome descritivo – outras informações (por exemplo, "LaserJet 5M in cer lab")

www.snow666.blogspot.com
Snow666

As próximas três linhas em nosso exemplo contêm configuração para o nome de dispositivo (`lp`), diretório de spool (`sd`) e arquivo de log de erros (`lf`). A última linha especifica uma conexão leitura-gravação com a impressora (`rw`), o tamanho máximo do arquivo (`mx`, ilimitado neste caso), o nome da máquina remota (`rm`) e o nome da impressora remota (`rp`).

Tarefas submetidas ao sistema de impressão sem um destino específico são direcionadas para a primeira impressora que tiver "lp" como um de seus nomes alternativos. Você não deve usar lp como nome primário de uma impressora; isto dificulta a mudança da impressora-padrão. Se nenhuma impressora tiver o nome lp, a primeira impressora no arquivo **printcap** será a impressora-padrão para todo o sistema.

Variáveis printcap

A flexibilidade do arquivo **printcap** é em grande parte responsável pela adaptabilidade do sistema de impressão LPD. Os detalhes são documentados na página de manual **printcap**, de modo que discutiremos somente as variáveis mais comuns aqui. Elas são mostradas na Tabela 24.2.

Tabela 24.2 Variáveis printcap comumente usadas.

Nome	Tipo	Significado	Exemplo
`sd`	string	Diretório de spool	`sd=/var/spool/lpd/howler-lw`
`lf`	string	Arquivo de log de erros	`lf=/var/log/lpd-errors`
`lp`	string	Nome de dispositivo	`lp=/dev/lp0`
`af`	string	Arquivo contábil	`af=/var/adm/lpr.acct`
`rm`	string	Nome da máquina remota	`rm=beast.xor.com`
`rp`	string	Nome da impressora remota	`rp=howler-lw`
`of`	string	Filtro de saída	`of=/usr/libexec/lpr/lpf`
`if`	string	Filtro de entrada	`if=/usr/sbin/stylascii`
`mx`	número	Tamanho máximo do arquivo	`mx#0`
`sh`	booleano	Suprime cabeçalhos	`sh`

Todas as entradas **printcap** devem incluir ao menos uma especificação do diretório spool (`sd`), do arquivo de log de erros (`lf`) e do dispositivo de impressão (`lp`). Se você tiver uma impressora moderna, deve especificar que ela seja aberta para leitura e gravação (`rw`), de modo que possa enviar mensagens de erro e de estado de volta para o host.

sd: diretório de spool

Cada impressora deve ter seu próprio diretório de spool. Todos os diretórios de spool devem estar no mesmo diretório-pai (usualmente **/var/spool/lpd**) e devem ter o mesmo nome que o nome completo da impressora que eles atendem (anchor-lj no exemplo anterior). É necessário um diretório de spool mesmo se a impressora que está sendo descrita residir numa máquina diferente; arquivos colocados em spool são armazenados localmente até poderem ser transmitidos ao sistema remoto para impressão.

Ao instalar uma impressora nova, você tem de criar manualmente seu diretório de spool. As permissões devem ser 775, com proprietário e grupo como daemon.

O diretório de spool para uma impressora também contém dois arquivos de estados: **status** e **lock**. O arquivo **status** contém uma descrição de uma linha do estado da impressora. Essas informações são mantidas por **lpd** e vistas com o comando **lpq**. O arquivo **lock** impede que várias chamadas de **lpd** se tornem ativas numa única fila e armazena informações sobre a tarefa ativa. As permissões sobre o arquivo **lock** são manipuladas por **lpc** para controlar o spooling e a impressão na impressora.

Veja o Capítulo 11 para obter maiores informações sobre arquivos de log.

lf: arquivo de log de erros

Erros gerados por filtros de impressão são colocados em log no arquivo citado nesta variável. Um log de erros pode ser compartilhado por todas as impressoras e ele pode ser colocado em qualquer lugar que você quiser. Quando é feita uma entrada de log, o nome da impressora problemática será incluído. Até mesmo impressoras remotas devem ter arquivos de log caso aconteça algum problema de comunicação com a máquina remota.

www.snow666.blogspot.com
Snow666

Tenha em mente que **lpd** envia mensagens de erro ao syslog. Alguns filtros também enviam suas mensagens de erro ao syslog, não deixando nada em seus arquivos de log especificado por **printcap**. Verifique esses dois locais quando surgir algum problema.

lp: nome de dispositivo
O nome de dispositivo para uma impressora tem de ser especificado caso a impressora seja local. Este nome é normalmente o arquivo contido no diretório **/dev** que representa a porta à qual a impressora está conectada. Se a entrada **printcap** endereça uma impressora de rede (isto é, uma impressora em sua LAN, não apenas uma impressora Linux "remota"; veja a página 555), a variável `lp` deve ser um ponteiro para um arquivo fictício. (Gostamos de usar **/var/spool/lpd/***impressora***/.null** para nossos arquivos fictícios.) Essa variável não precisa ser definida para impressoras remotas, mas se ela estiver definida, o arquivo tem de existir.

lpd usa um bloqueio de relatório no arquivo `lp` para determinar se a impressora está sendo usada. Mesmo que a impressora seja realmente acessada através de uma conexão de rede, você deve fornecer um valor para a variável `lp`. Especifique um arquivo exclusivo já existente num disco local.

rw: modo de abertura de dispositivo
Se uma impressora é capaz de enviar informações de estados de volta ao host por meio do seu arquivo de dispositivo, a variável booleana (`rw`) deve ser especificada para solicitar que o dispositivo seja aberto tanto para leitura quanto para gravação. O modo leitura-gravação é útil para fins de contabilidade e de relatórios de estados e alguns filtros o requerem.

af: arquivo contábil
Mesmo que você não pretenda cobrar pelo uso da impressora, a contabilidade de impressão pode lhe dar uma boa idéia sobre como os seus recursos de impressão estão sendo consumidos. Recomendamos que você habilite a contabilidade para todas as impressoras compartilhadas. Isso é feito simplesmente se especificando um arquivo contábil. Esse arquivo somente precisa ser especificado e estar na máquina com a qual a impressora está conectada fisicamente, já que os registros contábeis não são gravados antes de uma tarefa ser efetivamente impressa.

Por convenção, os arquivos de dados contábeis são normalmente chamados **/var/adm/impressora-acct**. Eles listam o número de páginas impressas para cada tarefa (usualmente uma declaração falsa), os nomes dos hosts em que as tarefas se originaram e os nomes de usuários dos proprietários das tarefas.

É responsabilidade do filtro de entrada da impressora gerar registros contábeis. Em impressoras PostScript, a menos que o filtro realmente consulte a impressora para saber a contagem de páginas antes e depois da tarefa ser encerrada, as contagens de páginas são extremamente suspeitas.

mx: limites de tamanho de arquivo
A variável `mx` configura um limite na quantidade de dados que podem ser colocados em spool de uma só vez. Entretanto, os tamanhos de arquivos são insignificantes para as impressoras, a não ser as impressoras de linha. Os arquivos Small Post-Script ou PCL poderiam imprimir centenas de páginas de lixo. Essa incoerência entre o tamanho do arquivo e o comprimento de página é particularmente evidente quando estudantes tentam imprimir as versões binárias compiladas de suas tarefas de programação.

Em alguns sistemas, `mx` usa como padrão algum valor diferente de 0 (nenhum limite), e uma entrada `mx#0` explícita é necessária para permitir grandes tarefas. Note que `mx` é um campo numérico; portanto, a entrada `mx=0` é incorreta.

Se você realmente precisa controlar o número de páginas que as pessoas podem imprimir, precisará usar filtros personalizados ou mudar para LPRng.

rm e rp: informações de acesso remoto
Na maioria das situações, você vai querer acessar uma impressora de mais de uma máquina da rede. Mesmo se a impressora for um dispositivo de rede, você deve escolher uma única máquina para ser a responsável pela comunicação com ela. Todas as demais máquinas devem encaminhar tarefas para o manipulador designado. Essa configuração permite que **lpd** cuide de colocar as tarefas em filas ordenadas em vez de ter várias máquinas constantemente brigando pelo controle da impressora. Isso também oferece um único local para procurar quando a impressão não estiver funcionando.

Máquinas remotas (máquinas que não estão diretamente conectadas à impressora) possuem uma única entrada **printcap** que informa para onde enviar a tarefa, como no exemplo da seção 24.6. A variável `rm` especifica a máquina para a qual as tarefas devem ser enviadas, e a variável `rp` dá o nome da impressora naquela máquina. Os detalhes sobre impressão remota são descritos na seção 24.6.

www.snow666.blogspot.com
Snow666

O fato de as entradas **printcap** serem diferentes para impressoras locais e remotas necessita de um pouco de subterfúgio por parte do administrador de sistemas se um arquivo **printcap** tiver de ser compartilhado entre várias máquinas. A solução é fazer com que os nomes local e remoto para uma impressora sejam distintos; por exemplo, howler-lw-local e howler-lw. Essa configuração torna howler-lw uma impressora "remota" mesmo na máquina onde ela reside efetivamente, mas isto é perfeitamente OK. Porém, você terá de se referir a howler-lw-local caso queira usar o comando **lpc**.

of, if, nf: filtros de impressão

Os filtros atendem a uma série de objetivos. O filtro de impressão padrão (normalmente **/usr/lib/lpf**) acomoda várias seqüências que não são de impressão e grava um registro contábil, caso seja apropriado. Infelizmente, os filtros não são padronizados. Qualquer um dos diversos pacotes de filtragem poderia executar a mesma tarefa, porém cada fabricante tende a ter filtros próprios.

Caso tenha uma impressora somente para impressão de caracteres, você realmente não precisa se preocupar com os filtros. Porém, se tiver uma impressora jato de tinta, laser ou PCL, talvez tenha de usar um filtro para realizar a conversão de formato apropriada.

Os filtros normalmente são apenas scripts de shell que chamam uma série de programas conversores. O programa de filtragem tem de aceitar a tarefa de impressão em entrada-padrão, converter a tarefa para um formato apropriado ao dispositivo e enviar o resultado à saída-padrão.

Se o usuário não especificar um filtro ao executar **lpr**, será usado `if` (input filter, filtro de entrada) ou `of` (output filter, filtro de saída). Os nomes são enganosos — na verdade ambos enviam dados a uma impressora.

Se a entrada **printcap** listar um filtro de entrada, mas não especificar um filtro de saída, o dispositivo será aberto uma vez para cada tarefa. A expectativa é a de que o filtro envie uma tarefa para a impressora e depois saia.

Ao contrário, se um filtro de saída for especificado sem um filtro de entrada, **lpd** abrirá o dispositivo somente uma vez e chamará o programa de filtragem somente uma vez, enviando todas as tarefas da fila num grande bloco. Essa convenção é perfeita para dispositivos que levam um bom tempo para se conectarem; porém, tais dispostivos são raros.

Se forem especificados tanto um filtro de entrada quanto um filtro de saída, a página banner será enviada ao filtro de saída (e o filtro de saída será chamado mesmo se banners estiverem desativados). O filtro de entrada será chamado para processar o restante da tarefa. Essa combinação de opções é realmente muito confusa para meros mortais. Evite-a. Use LPRng caso tenha necessidades complexas de filtragem.

Se tiver de criar filtros, tente usar filtros de entrada — eles são mais fáceis de serem depurados.

Os filtros de entrada são chamados com vários argumentos. Os mais interessantes deles são o nome de usuário, o host de origem e o nome do arquivo contábil. Se quiser realizar contabilizações para a impressora, o filtro de entrada terá de gerar os registros contábeis e anexá-los ao arquivo contábil. Se quiser restringir o acesso a uma impressora (para, por exemplo, negar impressão ao usuário "guest"), o filtro de entrada também terá de cuidar disso, já que **lpd** não possui nenhuma forma implícita para impedir que usuários individuais imprimam seus trabalhos.

Para esclarecer os empregos dos filtros, vamos dar uma olhada num exemplo simples de um filtro de entrada (`if`). O exemplo é para uma impressora PostScript conectada a uma linha serial na máquina local:

```
#!/bin/csh -f
/usr/local/bin/textps $* | /usr/local/bin/psreverse
```

Pelo fato de a impressora estar conectada serialmente, **lpd** cuida da abertura do dispositivo usando os modos corretos, conforme especificado em **/etc/printcap**. O primeiro programa chamado é **textps**, que procura a entrada e decide se ela é PostScript (que nossa impressora espera) e, em caso negativo, converte-a para PostScript. **textps** obtém todos os argumentos de filtragem que foram passados (o `$*`) e espera-se que ele gere registros contábeis com base nessas informações.

O segundo programa, **psreverse**, inverte a ordem das páginas de maneira que elas apareçam empilhadas apropriadamente.

Variáveis printcap para dispositivos seriais

As próximas variáveis **printcap** são úteis somente para impressoras seriais locais. Se estiver configurando uma impressora de rede, pule o restante desta seção. Caso contrário, abra o seu manual, procure as especificações de comunicação de sua impressora e faça a leitura.

Controlamos três tipos de configuração de comunicação através de **printcap**: a taxa de transmissão, os bits de "flag" e os bits de "modo local".

www.snow666.blogspot.com Snow666

br: taxa de transmissão

Se sua impressora estiver conectada a uma porta serial, precisará de uma entrada `br`. Uma impressora serial é como qualquer outro dispositivo de hardware: para operação correta, ela e seu computador host devem estar de acordo com um conjunto comum de parâmetros de comunicação, como velocidade, paridade e controle de fluxo. A configuração de uma impressora é bem parecida com a de um terminal. Veja o Capítulo 7 para obter informações genéricas sobre dispositivos seriais e cabeamento.

A taxa de transmissão é a velocidade na qual a comunicação ocorre (em bits por segundo) e é um inteiro simples. Já que ela é um valor numérico, você usa o símbolo # para configurá-lo. Por exemplo, `br#38400` configura a taxa de transmissão para 38.400 bps.

fc e fs: bits de flags

Ficar se sujando com os bits de flags normalmente é necessário apenas se você estiver tentando configurar uma velha impressora de impacto como um Teletype. Da mesma forma que os bits de modo local (a seguir), os bits de flag são inteiros, mas cada bit dentro do número modifica o comportamento da porta de maneira especial. Configurar corretamente estes parâmetros requer que você procure o significado de cada bit na página de manual **termios** (consulte a seção sobre constantes `c_cflag`) e adicione os valores para os bits que você deseja ativar ou não por meio de **/usr/include/bits/termios.h**. Os ajustes somente precisam ser especificados na máquina na qual a impressora está conectada.

Você pode atribuir duas variáveis ao ajustar os bits de flags: `fc` e `fs`. `fc` (flag clear) especifica os bits que devem ser desativados e `fs` (flag set) especifica aqueles que devem ser ativados. Os bits não atribuídos a nenhuma das variáveis assumem valores-padrão. É insignificativo definir um bit e depois limpá-lo.

A página de manual **termios** explica detalhadamente o significado de cada bit de flag. Os bits de flag podem parecer assustadores, mas desde que você saiba quais devem ser as configurações de comunicação de sua impressora, é realmente fácil configurar estes bits.

xc e xs: bits de modo local

Os bits de modo local são úteis somente para impressoras de linha seriais. As variáveis `xc` e `xs` ativam e desativam bits de modo individuais quase que da mesma maneira que `fc` e `fs` desativam e ativam os bits de flags. A diferença entre os dois conjuntos de bits é que os bits de modo local configuram o driver serial, ao passo que os bits de flags configuram o link de comunicação em si. A maioria dos bits de modo são destinados a uso em terminais de vídeo interativos e, portanto, não são relevantes para a configuração de impressoras.

Extensões de printcap

Um excelente recurso do sistema **lpr/lpd** é o fato de não importar se você fornecer valores para variáveis **printcap** não padronizadas. Freqüentemente, quando uma determinada impressora precisa de informações de configuração adicionais além das básicas que o sistema define, você pode colocar variáveis extras em **printcap** para que os filtros de impressão as usem.

Por exemplo, o filtro de saída para uma impressora de rede poderia precisar saber o nome de rede do dispositivo. A entrada **printcap** para a impressora poderia conter algo como

```
:nn=laser.colorado.edu:\
```

O emprego de extensões **printcap** permite que todas as informações de configuração para uma impressora sejam armazenadas num local conveniente. Se você vir variáveis no arquio **printcap** que não são discutidas na página de manual sobre **printcap**, verifique a documentação para filtros de impressão para obter os significados das variáveis.

Nossa instalação tira proveito desse recurso para documentar a localização física de cada impressora. Nossas impressoras possuem entradas como

```
:lo=Room 423, Engineering building:\
```

Temos scripts que monitoram a quantidade de toner e de papel nas impressoras e enviam e-mails para o pessoal de suporte com instruções como "leve mais papel para a sala 423 no prédio da Engenharia" quando necessário. Para maiores informações sobre monitoramento de dispositivos de rede, veja o Capítulo 20, *Gerenciamento e Depuração de Redes*.

Snow666

Imprimindo em algo que não seja uma impressora

Vimos recentemente um caso de "mau uso criativo" por Sean McCreary no qual o sistema de impressão LPD foi usado para colocar em spool música no formato MP3 para um software jukebox. No mínimo esse é um grande testemunho da flexibilidade do sistema de impressão.

A entrada printcap era parecida com o seguinte:

```
mp3-local:\
    :sd=/var/spool/lpd/mp3-local:\
    :lf=/var/log/lpd-errs:\
    :if=/usr/local/lib/mp3-play:\
    :lp=/dev/null:\
    :mx#0:
```

O verdadeiro MP3 player, **amp**, não lê por meio de stdin na situação-padrão, de modo que um script de uma linha chamado **mp3-play** o interfaceava com o sistema de impressão:

```
#!/bin/sh
exec /usr/local/bin/amp -
```

24.5 LPRng

LPRng é um spooler de impressão relativamente novo que se baseia no sistema LPD e é o sistema de impressão padrão que vem com as distribuições Red Hat e Debian. Atualmente mantidos por Patrick Powell da AStArt Technologies, LPRng combina de forma bem-sucedida os melhores recursos dos esquemas de impressão dos tradicionais Berkeley e UNIX System V.

Um dos maiores problemas com o sistema de impressão LPD é a necessidade da maior parte do software de impressão ser executada como root. Além de fazer como que clientes **lpr** rodem setuid para root, os filtros LPD também são executados como root. Pelo fato de os filtros normalmente serem scripts de shell, trata-se de um ponto de risco.

LPRng soluciona o problema permitindo que clientes sejam executados como usuários comuns. Nos casos em que LPRng não tiver de interagir com clientes não-LPRng, até mesmo o daemon de impressão poderá ser executado como usuário não-root. O pacote também acrescenta um bocado de recursos novos em termos de segurança que faltavam na maioria dos sistemas LPD.

Um dos recursos mais úteis de LPRng é sua capacidade de produzir diagnósticos completos e mensagens de erros. Em vez de falhar silenciosamente ou retornar uma mensagem criptográfica, os programas neste pacote dão explicações bem descritivas e úteis do que está dando errado.[5]

Embora o LPD suporte o controle de acesso através do arquivo **/etc/hosts.lpd**, não há nenhuma autenticação disponível. LPRng suporta métodos de autenticação Kerberos 5, SSL e PGP.

Finalmente, LPRng oferece alguns excelentes recursos de gerenciamento de filas, a maioria dos quais é proveniente do UNIX System V. LPRng inclui redirecionamento dinâmico de filas de impressão, suporte a várias impressoras numa fila e até mesmo balanceamento de cargas entre várias impressoras.

É fácil instalar LPRng em distribuições Linux que não o incluam por padrão. Mesmo que você não queira instalar LPRng em todos os pontos de sua rede, considere o emprego em seu servidor de impressão. Essa etapa simples lhe trará grande parte da segurança e funcionalidade de um emprego para toda a rede com muito pouco trabalho.

Os comandos LPRng

A versão LPRng de **lpr** mantém compatibilidade com versões anteriores da maioria das demais implementações **lpr**. A única exceção é a opção **-s**, que é ignorada silenciosamente pelo **lpr** de LPRng. Essa opção era usada originalmente para criar um link simbólico para o arquivo a ser impresso em vez de criar uma cópia, que era útil para imprimir arquivos grandes.

A versão LPRng de **lpr** fornece uma série de novos recursos. Particularmente úteis são a opção "verbose" (prolixo) (**-V**) e a opção de depuração (**-D5** para o mais detalhado).

Os comandos do LPRng permitem que especifiquemos o host e a porta da impressora além de seu nome. Esse recurso permite que usuários imprimam em impressoras remotas que não são definidas no arquivo **printcap** local.

Para especificar ao qual host a impressora está conectada, anexe *@nomedohost* ao nome da impressora. Uma porta é especificada anexando-se *%porta*. Por exemplo:

```
% lpr -Phowler-lw@beast 8552 nome-do-arquivo
```

5. Talvez o comitê de padrões Linux estivesse de férias no dia em que o LPRng veio à tona para revisão ...

Snow666

Esse método de especificar uma impressora funciona para as versões LPRng de **lpr**, **lpq**, **lprm** e **lpc**, o que significa que clientes de impressão normalmente não precisam de um arquivo **printcap**.

Se não for especificada uma impressora na linha de comandos, é usado o valor da variável de ambiente PRINTER. Se a variável de ambiente PRINTER não estiver configurada é usada a primeira entrada do arquivo **/etc/printcap**. Caso não exista nenhum arquivo **printcap**, é usada a impressora-padrão especificada no arquivo **lpd.conf**.

LPRng adiciona vários recursos novos a **lpq**. Vários formatos de saída alternativos estão disponíveis através das opções curto (**-s**), longo (**-l**) e detalhado (**-L**). Essas opções podem ser úteis com scripts que dependem da saída de **lpq**. O comando aceita agora uma opção para realizar consultas periódicas (**-t** *segundos*). Finalmente, a opção de depuração descrita é suportada. Com a opção **-D5**, a saída do comando incluirá mensagens de estado detalhadas de impressoras locais e remotas e de spoolers de impressão.

Talvez o melhor acréscimo ao **lpc** de LPRng é a sua capacidade de priorizar entradas de filas baseadas em algo mais útil do que a ordem na qual eles foram submetidos. As regras para prioridade de filas podem ser complexas o bastante para acomodarem qualquer política desejada. Diferentemente do **lpc** tradicional, o **lpc** de LPRng pode ser executado via rede.

O **lprm** de LPRng permite a eliminação de tarefas de acordo com algum dos vários critérios existentes. No sistema tradicional, o ID de tarefa ou nome de usuário era exigido. Com LPRng, as tarefas também podem ser removidas de um spool por uma expressão regular que é comparada ao ID de tarefa.

/etc/lpd.conf: configurar lpd

O servidor **lpd** de LPRng é altamente configurável por meio do arquivo **/etc/lpd.conf** file. De fato, na época em que este livro era escrito, 185 parâmetros diferentes podiam ser configurados nesse arquivo. A maioria dos parâmetros configura valores-padrão de sistemas operacionais e de diretórios. Também está disponível um mecanismo pelo qual podemos fornecer valores-padrão para variáveis **/etc/printcap** não especificadas.

A melhor maneira de criar um arquivo **lpd.conf** para a sua instalação é copiar o arquivo **lpd.conf** na raiz da distribuição LPRng. Esse arquivo contém uma descrição, opção por opção, com exemplos. Caso não tenha tempo suficiente para ficar navegando por essa longa lista de opções, consulte a página de manual on-line para **lpd.conf**, que contém explicações de todas as configurações e até mesmo um exemplo.

/etc/lpd.perms: configurar controle de acesso

Com **/etc/lpd.perms**, você pode estabelecer políticas de impressão muito complexas. Regras para controlar o acesso à impressão são aplicadas a tarefas de impressão na ordem na qual elas aprecem no arquivo **lpd.perms**.

Regras são constituídas por duas partes. A primeira delas é simplesmente um identificador `ACCEPT` ou `REJECT` que determina se a operação especificada será permitida. Ela é seguida por uma série de estruturas que especificam a que pessoas, hosts, impressoras ou operações a regra se aplica.

Por exemplo, a linha a seguir informa ao sistema de impressão que o usuário evi no host remoto beast pode imprimir tarefas, colocar tarefas em spool, remover tarefas e verificar o estado das tarefas na impressora howler-lw.

```
ACCEPT SERVICE=P,R,M,Q REMOTEHOST=beast REMOTEUSER=evi
   PRINTER=howler-lw
```

Os códigos de uma letra na estrutura `SERVICE` determinam a que operações a regra está se referindo. A Tabela 24.3 mostra os valores que a estrutura `SERVICE` pode conter.

Tabela 24.3 Códigos SERVICE em /etc/lpd.perms.

Código	Ação	A que ele se refere
P	Impressão	Imprime uma tarefa de uma fila
C	Controle	Usa **lpc** (permite que secretarias desbloqueiem a impressora)
R	Spool	Submete tarefas à fila de spool com **lpr**
M	Remoção	Remove tarefas da fila com **lprm**
Q	Estado	Obtém informações de estado com **lpq**
X	Conexão	Conecta-se à **lpd**

Snow666

A melhor maneira de configurar o arquivo **lpd.perms** de sua instalação é modificar a versão de exemplo inclusa com o pacote (instalado em **/etc**). Esse arquivo descreve detalhadamente as linhas `ACCEPT` e `REJECT` e inclui vários exemplos esclarecedores. Todas as possíveis opções de configuração são descritas nesse arquivo.

Configurando o arquivo printcap

A coisa mais importante para se lembrar em relação ao arquivo **printcap** de LPRng é que ele é totalmente compatível com versões anteriores do arquivo **printcap** do LPD tradicional. Isto é, todos os arquivos **printcap** tradicionais também são arquivos **printcap** de LPRng válidos.

Entretanto, o LPRng também fornece muitos recursos **printcap** novos. As linhas podem ser continuadas simplesmente por recuo em vez de uma barra invertida anexada. Os nomes de variáveis não são mais limitados a dois caracteres. Várias macros estão disponíveis para ajudar a simplificar a configuração. As entradas podem ser destinadas a determinados hosts.

Provavelmente a ferramenta de configuração mais interessante que o LPRng nos traz seja o programa **checkpc**. **checkpc** verifica a validade e a consistência do arquivo **printcap**. Ele alerta sobre a falta de diretórios de spool, arquivos faltantes e permissões incorretas. Se você executar **checkpc** com a opção **-f**, ele tentará solucionar problemas simples (como criar um diretório ou alterar as permissões de um arquivo). Executar **checkpc** com a opção **-D5** gera um diagnóstico completo.

Filtros

Além de fornecer os recursos de filtragem do LPD, o LPRng permite que sejam aplicados filtros a tarefas enviadas a impressoras remotas. Esse recurso permite formatação de dados personalizados para um host em vez de uma impressora.

As variáveis **printcap** de filtro de saída (`of`) e de filtro de entrada (`if`) atuam de forma ligeiramente distinta no LPRng. O filtro de saída sob **lpr** do LPD tradicional é aplicado tanto ao documento quanto ao banner. No LPRng, o filtro de saída é usado somente para o banner. O filtro de entrada de LPRng é usado somente para imprimir texto. Filtros para outros tipos de arquivos são especificados com tags chamadas *x*`f`, onde *x* é uma abreviatura, de uma letra, do tipo. Você pode desabilitar completamente os filtros para uma tarefa adicionando **-Y** para a linha de comando **lpr**.

Inclusos no LPRng estão os comandos **lpbanner**, **pclbanner** e **psbanner**. Esses comandos criam banners, respectivamente, de texto, PCL e PostScript. Cada um deles aceita argumentos na linha de comando que especificam o nome do banner, o nome de login e o título da tarefa.

O LPRng vem com um filtro particularmente útil, **ifhp**, que imprime para uma variedade tremenda de impressoras. Embora seja desenhado primariamente para suportar impressoras HP, ele é capaz de trabalhar com a maioria das impressoras sem necessidade de qualquer modificação. As impressoras que não são suportadas internamente em geral são complicadas para serem colocadas em operação.

Contabilidade

Nos dias das impressoras de linha, a contabilidade era uma tarefa simples. Era fácil obter-se uma contagem precisa do número de páginas impressas de qualquer tarefa contando-se as linhas. Entretanto, esse método falha completamente em arquivos PostScript e em outros tipos de arquivos.

A única maneira precisa de contabilizar o número de páginas usadas por uma tarefa é realmente contar estas páginas à medida que elas saem da impressora. Felizmente, muitas impressoras modernas possuem um contador de número de páginas que elas já imprimiram ao longo de suas vidas úteis. Verificando-se esse contador antes e depois de uma tarefa de impressão, o driver da impressora pode determinar o verdadeiro número de páginas.

24.6 ADICIONANDO UMA IMPRESSORA

Nesta seção discutimos o processo global de configuração de uma impressora. Examinamos várias situações:

- Configuração de uma impressora serial ou paralela local
- Aceitação de tarefas de impressão **lpd** da rede
- Impressão num servidor de impressão **lpd** de rede

Nas discussões a seguir, partimos do pressuposto de que o hardware da impressora já tenha sido fisicamente conectado a um host ou à rede. Veja o Capítulo 7, *Dispositivos Seriais*, para obter informações sobre como conectar impressoras seriais, e o Capítulo 15, *Hardware de Rede*, para informações genéricas sobre conexão de dispositivos a uma rede.

Snow666

As impressoras de rede exigem um pouco mais de configuração. Em particular, devem ser atribuídas a elas endereço IP. Isso é normalmente feito através de um de dois métodos.

Consulte a seção 13.9, para obter maiores informações sobre o DHCP.

Primeiramente, a maioria das impressoras moderna tem o recurso de serem inicializadas pela rede por meio de um servidor DHCP ou BOOTP. Esse método funciona bem em ambientes com várias impressoras homogêneas. Alternativamente, todas as impressoras de rede permitem que se configurem seus endereços IP pela console.

Algumas vezes a "console" é uma porta serial, mas normalmente ela é apenas um esquema para usar botões do painel da impressora. Após sua impressora estar na rede, você pode enviar um ping a ela, certificando-se de assegurá-la conforme discutido na seção 24.9.

Ferramentas disponíveis para Linux podem ajudá-lo a configurar outros recursos de impressão, como bandejas de papel. Uma dessas ferramentas é o XPP, disponível em cups.sourceforge.net/xpp.

Para que as próximas seções fluam mais facilmente, tratamos de um conjunto de hardware consistente. O servidor usa um spooler **lpd** genérico conectado a uma impressora laser PostScript chamada howler-lw.

Configurando uma impressora local

Para cada impressora local, você precisará criar uma entrada em **/etc/printcap**. Uma entrada **printcap** se parece tipicamente com o seguinte:

```
howler-lw|howl|laserjet:\
   :sd=/var/spool/lpd/howler-lw:\
   :mx#0:\
   :lp=/dev/lp0:\
   :sh:
```

Essa entrada define três nomes para a impressora, especifica um diretório de spool e uma porta, elimina o tamanho máximo de tarefas e suprime cabeçalhos de impressão.

Podemos deduzir pelo arquivo de dispositivos que se trata de uma impressora paralela. As impressoras seriais são configuradas de forma similar. O dispositivo seria provavelmente **/dev/ttyS0** (ou **S1** para a sua segunda porta serial) em vez de **/dev/lp0**, e parâmetros como taxa de transmissão provavelmente teriam de ser especificados. Procure as configurações relevantes na página de manual **printcap**.

Aceitando tarefas de impressão da rede

Assim que tiver conectado uma impressora localmente, você poderá fazer impressões nela de outras máquinas da rede. Primeiramente, certifique-se de que é capaz de imprimir localmente sem qualquer dificuldade. A seguir, para o seu arquivo **/etc/hosts.lpd**, acrescente os nomes das máquinas-cliente das quais você quer aceitar imprimir tarefas. Veja a próxima seção para a configuração no cliente.

Imprimindo num servidor de impressão da rede

Como acontece com todos os sistemas de impressão LPD, novas impressoras de rede precisam de uma entrada **/etc/printcap** na máquina-cliente. Para nossa LaserJet, adicionamos algo como se segue:

```
howler-lw|lp|8-6|"LaserJet 5M, called howler-lw on beast":
   :lp=/var/spool/lpd/howler-lw/.null:\
   :rm=beast:rp=howler-lw:\
   :sd=/var/spool/lpd/howler-lw:mx#0:
```

A seguir criaríamos o diretório de spool e o arquivo **.null** no cliente:

```
# mkdir /var/spool/lpd/howler-lw
# touch /var/spool/lpd/howler-lw/.null
# chown -R daemon /var/spool/lpd/howler-lw
# chgrp -R daemon /var/spool/lpd/howler-lw
# chmod 775 /var/spool/lpd/howler-lw
```

Se o servidor **lpd** for uma máquina de verdade (e não apenas uma impressora inteligente), certifique-se de poder imprimir tarefas nesta máquina antes de configurar clientes de rede.

Para testar a impressão do cliente, use uma seqüência de comandos similar à seguinte:

```
# lpc start howler-lw
# lpr -Phowler-lw /etc/motd
# lpq -Phowler-lw
```

www.snow666.blogspot.com
Snow666

Detalhes específicos a cada distribuição

A configuração de impressoras é uma área na qual as variantes do Linux diferem substancialmente. Sempre que possível, provavelmente é mais indicado usar as ferramentas de configuração com interface gráfica com o usuário fornecidas pelo fabricante para configurar impressora.

O sistema de impressão padrão do Red Hat é uma implementação bem comum do LPRng. Algumas das ferramentas do Red Hat são muito úteis, especialmente o **printtool** gráfico que automatiza a edição do arquivo **/etc/printcap** e o script **lpunlock**, que recupera servidores de impressão bloqueados. O amado **printtool** o segura pela mão durante a configuração de uma variedade de tipos de impressoras, inclusive impressoras LPD, CIFS (Windows) e NetWare (NCP) locais e remotas. Esteja atento que o **printtool** requer que o formato do arquivo **printcap** seja muito preciso; se você editar o arquivo manualmente, talvez não seja capaz de abri-lo mais no **printtool**. Definições de impressoras locais têm de ser colocadas em **/etc/printcap.local**; elas são transferidas automaticamente para **/etc/printcap** quando **lpd** é iniciado. **checkpc -f** cria diretórios de spool de impressão para você.

O sistema de impressão padrão que vem com o SuSE é a versão BSD do LPD. O daemon **lpd** tem de ser habilitado incluindo-se a linha `START_LPD=yes` no arquivo **/etc/rc.config** antes da impressão poder funcionar. YaST, a ferramenta gráfica do SuSE para administração de redes, inclui funções que gerarão entradas **/etc/printcap** extremamente básicas para você. Porém, é provável que você precisará refinar ainda mais essas entradas editando manualmente o arquivo **printcap**.

O Debian usa como padrão o pacote LPRng sem nada de extra. Você terá de editar diretamente os arquivos de configuração (como **/etc/printcap**) para acrescentar ou alterar as configurações de impressão (ou fazer o download do configurador gráfico **printtool** usando **apt-get**).

24.7 Depurando Problemas de Impressão

As tarefas de impressão **lpd** de rede são entregues na porta TCP 515. A menos que você queira imprimir tarefas para estranhos, seu firewall deve bloquear todo o tráfego para esta porta proveniente da Internet. Para testar sua conectividade a um servidor **lpd** remoto, envie um **telnet** do cliente para a porta 515 do servidor. Se você conseguir estabelecer uma conexão, poderá ao menos verificar que a rede está funcionando e que **lpd** está rodando no servidor.

Caso tenha problemas ao depurar uma conexão de impressora remota, precisará procurar em seis (isto mesmo, em seis) lugares para descobrir o problema:

- O arquivo de log do sistema (conforme especificado em **/etc/syslog.conf**) na máquina que está hospedando a impressora, em busca de mensagens sobre problemas de permissão
- Os arquivos de log do sistema na máquina remetente, em busca de problemas relacionados à permissão e à determinação de nomes
- O arquivo de log do daemon de impressão no servidor, em busca de mensagens sobre nomes de dispositivos errados, formatos incorretos etc. (conforme especificado em **/etc/syslog.conf** para o recurso de syslog "lpr")
- O arquivo de log do daemon de impressão na máquina remetente, em busca de filtros faltantes, impressoras desconhecidas, diretórios faltantes etc.
- O arquivo de log de impressão na máquina que faz a impressão, em busca de erros na transmissão de uma tarefa (conforme especificado pela variável `lf` no arquivo **/etc/printcap**)
- O arquivo de log de impressão na máquina remetente, em busca de erros sobre pré-processamento ou colocação em fila da tarefa

Ao fazer a depuração em impressoras remotas, sempre tenha em mente que deve haver uma fila para a tarefa na máquina solicitante, uma maneira de decidir para onde enviar a tarefa e um método de enviar a tarefa à maquina remota. Na máquina que faz a impressão, deve haver um lugar para colocar a tarefa em fila, permissões suficientes para permitir que a tarefa seja impressa e uma maneira de gerar saída para o dispositivo.

Antes de começar a rastrear um problema de impressão em rede, certifique-se de poder imprimir da máquina que efetivamente abriga a impressora. Pode ser que na verdade você não tenha nenhum problema de rede.

Snow666

24.8 Software Comum para Impressão

Embora tanto LPD quanto LPRng disponham de sistemas adequados para colocação em filas, monitoramento e geração da saída de tarefas de impressão, nenhum deles provê muito da conversão de formatos necessária para operar impressoras modernas. Ferramentas adicionais que residem sobre o sistema de impressão podem fornecer esses recursos.

Nosso objetivo nesta seção não é o de lhe dizer tudo o que você precisa saber sobre estes pacotes, mas sim para você tomar conhecimento sobre a funcionalidade que eles anunciam.

ghostscript

ghostscript é um interpretador PostScript de distribuição gratuita que lhe permite ver arquivos PostScript em sua tela. Ele também é bom para comandar dispositivos de saída rasterizados (como WinPrinters baratas). Mesmo que sua impressora em particular não seja suportada diretamente, você poderá usar **ghostscript** como ponto de partida para criar o seu próprio driver. Esteja ciente de que criar um driver pode ser um processo muito complicado.

Várias versões diferentes de **ghostscript** estão disponíveis; vá ao site www.ghostscript.com para obter maiores informações sobre as diferenças entre elas. O Red Hat e o SuSE vêm com a versão ligeiramente comercial de **ghostscript**, o Aladdin (www.aladdin.com), que contém um bom nível de suporte para WinPrinters.

mpage

mpage é um conversor texto-para-PostScript que coloca várias páginas lógicas numa única página física. Esse truque é um grande preservador das árvores quando imprimimos coisas como o código-fonte em que não é necessário empregar fontes grandes e margens extensas.

enscript

A Adobe desenvolveu originalmente um produto chamado **enscript** para converter arquivos de texto em Post-Script para impressão. Este software, ou algo do gênero, faz com que a saída de texto tenha uma boa apresentação em impressoras PostScript. **enscript** dispõe de recursos de formatação de páginas como cabeçalhos de páginas ornamentados e impressão "dois em um" (duas páginas impressas com a metade do tamanho numa única folha).

Embora a versão Adobe deste programa não seja mais mantida, existe uma boa alternativa. O **enscript** do GNU é uma versão gratuita e de fonte aberto do **enscript** que é completamente compatível com a versão do Adobe. O **enscript** do GNU oferece uma ampla gama de recursos, inclusive destaque sensível à linguagem, suporte para vários tamanhos de papel, download de fontes PostScript e cabeçalhos definidos pelo usuário.

Você pode fazer o download do **enscript** do GNU na Web, no endereço, people.ssh.fi/mtr/genscript. Já que a maior parte do trabalho neste projeto é feito na Finlândia por Markku Rossi, o tipo de papel padrão é A4. O papel A4 é ligeiramente mais longo do que o carta, suficiente o bastante para cortar pedaços vitais de seus documentos como cabeçalhos, rodapés ou números das páginas. Para americanizar o **enscript** GNU, execute o comando **configure** com o seguinte argumento quando estiver instalando-o pela primeira vez:

```
# ./configure --with-media=Letter
```

O tipo de papel também pode ser configurado na linha de comando de **enscript**. Caso tenha se esquecido de alterar o tipo de mídia e usar **enscript** para imprimir numa impressora high-end que reconheça tipos de papel que ele tem carregado, pode ser que a impressora se recuse a efetuar a impressão.

24.9 Filosofia de Impressão

As principais coisas que se deve esperar ao lidar com impressoras são problemas e frustrações. Se tudo isso falhar, fique apenas contente por não ser MS-DOS.

Contabilidade do uso de impressoras

Você deve habilitar a contabilidade de impressoras mesmo que não pretenda cobrar pelo uso delas. O trabalho adicional é muito leve e você poderá ver exatamente quem está usando a impressora. A contabilidade de impressoras também dá excelente informações sobre as diversas fontes de tarefas de impressão, que são informações interessantes de ter quando estiver colocando impressoras novas.

Use páginas-banner somente quando necessário

O sistema de impressão pode enviar uma página de prefácio para cada tarefa mostrando o título da tarefa e o usuário que a submeteu. Essa página-banner pode ser útil em impressoras usadas por várias pessoas diferentes, mas é uma

Snow666

perda de tempo e de papel na maioria dos casos. Se não precisar das páginas-banner, suprima-as configurando a variável booleana **printcap**, sh.

Forneça lixeiras

Todos os tipos de papel para computador são recicláveis. Você pode usar as caixas em que o papel vem como lixeiras recicláveis. Coloque um aviso dizendo que nenhum material estranho (como grampos, clipes e jornais) deve ser jogado ali.

Forneça pré-visualizadores

Os usuários normalmente imprimem um documento, localizam um pequeno erro na formatação e acabam imprimindo novamente todo o trabalho. Esse desperdício de papel pode ser facilmente evitado com software que permita aos usuários visualizarem seus trabalhos na tela antes de serem efetivamente impressos.

O recurso de pré-visualização já está embutido na maioria dos editores WYSIWYG modernos, mas se seus usuários forem viciados em algum sistema de digitação mais antigo, você precisará providenciar alguma outra maneira de pré-visualizar documentos.

Para documentos PostScript esporádicos, você pode usar **ghostscript**; para **roff**, canalize a saída de **groff** para **ghostview**. Depois de ter fornecido os pré-visualizadores necessários, treine seus usuários para usá-los. Um bom emprego dos registros contábeis é a verificação de casos nos quais o mesmo documento vem sendo impresso repetidamente.

Compre impressoras baratas

A tecnologia de impressão é madura. Você não precisa gastar muito para obter excelente qualidade de impressão e mecânica confiável.

Não esbanje numa impressora de "grupo de trabalho" cara a menos que você realmente precise de uma. Não há nenhuma diferença de qualidade, e uma impressora "pessoal" média pode ser tão rápida e tão confiável, sem mencionar nas dezenas de quilos mais leve. Uma impressora de 10 páginas por minuto pode atender cerca de cinco usuários em tempo integral. Na maioria dos casos, é melhor comprar cinco impressoras de US$ 500 para um grupo de 25 usuários do que uma impressora de US$ 2.500.

Em geral você não deve comprar uma impressora (ou um disco rígido ou memória) de um fabricante de computadores. Suas impressoras normalmente provêm de outros fabricantes famosos, mas custando o dobro do preço. O melhor é investir em impressoras PostScript fabricadas para os mercados de PCs e Macintoshes. Particularmente, temos tendo boa sorte com as impressoras laser da HP e da Lexmark. Elas são produtos de alta qualidade capazes de jogar em spool tarefas de impressão **lpd** de rede e funcionarem bem com os drivers PostScript genéricos do Linux.

Assegure sua impressora

A maioria das impressoras mais novas suporta alguma forma de gerenciamento remoto. O gerenciamento remoto é bom para administradores de sistemas, pois permite uma configuração conveniente sem andar muito. Algumas maneiras comuns de acessar remotamente uma impressora são: **telnet**, HTTP e SNMP. Através da interface remota, você pode configurar o endereço IP, gateway-padrão, servidor de syslog, nome de comunidade SNMP, opções de protocolo e, mais importante ainda, senhas para as impressoras.

Como padrão, a maioria das impressoras administráveis remotamente é desprotegida e devem ter uma senha (ou quem sabe um nome de comunidade SNMP "ativo") atribuída como parte do processo de instalação. Consulte os manuais de instalação do fabricante de sua impressora para aprender a fazer isto em sua impressora em particular. Alguns fabricantes somente distribuem configuradores baseados em Windows, mas você poderá configurar a maioria das modernas impressoras de rede HP usando um navegador Web.

24.10 Exercícios

E24.1 Como você ativaria a contabilidade de impressão para o LPD? Por que a contabilidade de impressão é importante?

E24.2 Que etapas seriam necessárias antes de levar uma impressora para manutenção?
Usando o seu procedimento, o que acontece a tarefas de impressão colocadas em fila depois de a impressora ser trazida de volta?

E24.3 Leia a página de manual **hosts.equiv**. Note que o **hosts.lpd** possui o mesmo formato, exceto pelo fato de não oferecer suporte ao parâmetro *nomedeusuário*. Crie o seu próprio arquivo **hosts.lpd** que rejeita explicitamente todas as tarefas de impressão da máquina "rouge" e aceita as da máquina "doc-print".

Snow666

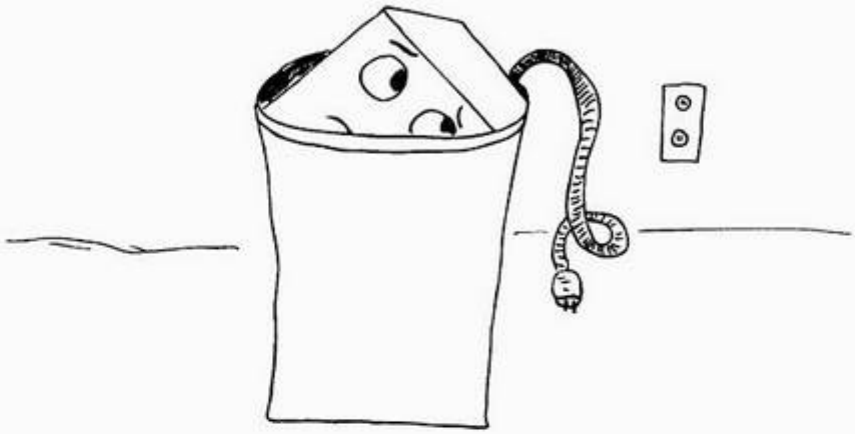
E24.4 Ao usar uma impressora conectada à rede capaz de colocar em spool seus próprios arquivos, por que ainda é uma boa idéia usar um servidor de impressão para controlar todas as tarefas de impressão?

★ **E24.5** Liste as etapas necessárias para adicionar uma nova impressora serial chamada "slow-bro" ao seu sistema Linux. A taxa de transmissão da impressora é 9600 e ela está conectada à segunda porta serial de um PC.

★★ **E24.6** Escolha um modelo atual de impressora (por exemplo, uma HP LaserJet 6MP) e investigue como suportá-lo no Linux. Você precisa configurar filtros? Você precisa de **ghostscript**? Ou a impressora não é suportada de forma alguma? Se a impressora for suportada, dê uma breve visão geral das etapas que seriam necessárias para instalá-la. Você ganhará pontos se o preço de varejo da impressora escolhida estiver abaixo de US$ 200.

Snow666

25 Manutenção e Ambiente



Era uma vez, dezenas de programadores que compartilhavam um único sistema, tal como o agora famoso VAX. Naquela época, uma tremenda quantidade de esforço era aplicada na manutenção de equipamentos e no fornecimento de um ambiente favorável ao seu crescimento. Administradores de sistemas tarimbados muitas vezes sabem tanto sobre sistemas de refrigeração baseados em etilenoglicol quanto sabem sobre gerenciamento de contas.

Os anos 90 trouxeram consigo um afluxo de estações de trabalho desktop e um afastamento das plataformas computacionais "grandes armários" (mainframes). Por um instante, parecia que os dias do CPD baseado numa máquina centralizada estavam contados. Recentemente, o paradigma cliente/servidor resultou numa dependência cada vez maior de plataformas servidoras rodando num sistema operacional que fornecesse flexibilidade, confiabilidade, segurança e desempenho. O Linux e o UNIX começaram a ocupar essa fatia de mercado e, como resultado, hordas de servidores se transferiram para essas salas de máquinas uma vez abandonadas (embora, em muitos casos, as salas de máquinas em si tenham sofrido um processo de downsizing — diminuição de tamanho). Prover um ambiente saudável e com boa manutenção para esses servidores é mais importante do que nunca.

Este capítulo dá algumas dicas sobre a manipulação e a manutenção de hardware, bem como dar a ele uma boa moradia.

25.1 Fundamentos sobre Manutenção

A manutenção de hardware era tradicionalmente algo coberto por um caro contrato anual de manutenção. Embora tais contratos ainda estejam prontamente disponíveis, hoje em dia é mais comum usar a abordagem "adote uma postura por conta própria" para manutenção de hardware.

Consulte a seção 29.11 para obter maiores informações sobre hardware em fase de aposentadoria.

Se você mantiver um livro de registros, uma rápida olhada em seus registros nos últimos seis a doze meses lhe dará uma idéia das taxas de falhas. É uma boa idéia manter um registro cuidadoso das falhas e substituições para que você possa avaliar de forma acurada as diversas opções de manutenção que você tem à disposição. Algumas peças dão problema com mais freqüência do que o previsto pelo fabricante, de modo que contratos de manutenção são algumas vezes não somente convenientes, como também financeiramente vantajosos. Mas lembre-se, chegará uma época quando todo o hardware deverá ser substituído e não mais mantido. Conheça o seu hardware e deixe-o ir graciosamente quando o tempo dele finalmente tiver chegado. Você até poderia considerar a doação de equipamentos obsoletos a uma escola ou universidade local. Para eles, raramente os equipamentos são tão antigos assim para não serem úteis.

Snow666

O melhor esquema de manutenção provavelmente seja o de estratégia da "garantia seletiva". Os fabricantes de unidades de disco oferecem garantias de até cinco anos de duração, e alguns módulos de memória até vêm com uma garantia por toda a vida. Muitas estações de trabalho possuem pelo menos um ano de garantia. Ao adquirir equipamento novo, pesquise em busca da melhor garantia — isto lhe poupará dinheiro a longo prazo.

25.2 Contratos de Manutenção

Vários dos principais fornecedores oferecem manutenção de hardware para equipamentos computacionais que eles próprios não vendem. Esses fornecedores normalmente estão ávidos para desbancar o fabricante original e se plantam na sua porta, por assim dizer. Algumas vezes você pode negociar contratos de manutenção muito atraentes colocando um fabricante contra um terceiro (fornecedor). Se possível, obtenha referências sobre todos os fornecedores de manutenção em potencial, preferivelmente de pessoas que você conheça e confie.

Manutenção local

Se você tiver um contrato de manutenção local, um técnico de serviços trará peças de reposição diretamente para onde se encontra sua máquina. O tempo de atendimento garantido varia de 4 a 24 horas; normalmente está dito explicitamente no contrato. Os tempos de atendimento durante o horário comercial podem ser menores do que em ouros períodos da semana.

Se estiver considerando um contrato de manutenção com rápido tempo de atendimento, normalmente vale a pena calcular o custo de manter à mão alguns sistemas de backup completos para que você possa rapidamente substituir computadores com problemas. Uma troca total de sistema lhe dará um tempo de reparação mais rápido do que o mais sofisticado dos contratos de manutenção, e, com os baixos preços do hardware hoje em dia, muitas vezes o investimento é mínimo.

Manutenção para troca de placas

Um programa para troca de placas requer que você e a sua equipe diagnostiquem problemas, talvez com a ajuda do pessoal de suporte de seu fabricante. Após os diagnósticos, você liga para um número de manutenção, descreve o problema e faz o pedido da(s) placa(s) que precisa(m) ser substituída(s). Normalmente elas são despachadas imediatamente e chegam às suas mãos no dia seguinte. Você então instala a placa, faz com que o hardware retorne a funcionar e devolve a placa defeituosa na mesma caixa em que veio a placa nova.

O fabricante normalmente irá querer atribuir um número RMA ("Return Merchandise Authorization", autorização de retorno de mercadoria) à transação. Certifique-se de escrever esse número nos documentos para despacho ao devolver a placa defeituosa.

Garantias

A extensão da garantia do fabricante deve desempenhar um papel significativo em seu cálculo do custo de propriedade num ciclo de vida de uma máquina. Uma garantia de um ano é padrão para computadores, mas garantias de vários anos ou mais não são raras.

Num ambiente universitário, parece ser mais fácil obter fundos governamentais para equipamentos de primeira linha do que para pessoal de suporte ou manutenção. Ocasionalmente pagamos por uma opção de "garantia estendida" em hardware novo (que também poderia ser descrita como manutenção pré-paga) para converter dinheiro destinado a equipamentos em dinheiro destinado à manutenção.

Se você adquirir as peças de um sistema computacional importante de vários fornecedores, as peças não chegarão necessariamente ao mesmo tempo. Com tantas peças de hardware, os maiores problemas de manutenção e de confiabilidade ocorrem quase sempre logo após a instalação. Falhas de hardware que ocorrem dentro de um a dois dias da instalação são conhecidas como "mortalidade infantil".

25.3 Cuidados na Manipulação de Placas

Os sistemas baseados em PC parecem exigir pelo menos quatro ou cinco placas adicionais para atingirem padrões de estação de trabalho (SCSI, som, vídeo, rede, memória... Hmm, afinal de contas o que tem na placa-mãe). Placas de circuito devem ser manipuladas com cuidado, não podem cair no chão, não podem ter café derrubado sobre elas, nem livros empilhados sobre elas etc. A maioria dos engenheiros de campo (aquele pessoal de reparos que vêm com o seu contrato de manutenção) são dez vezes mais bruscos com as placas do que parece ser razoável.

www.snow666.blogspot.com
Snow666

Eletricidade estática

Os componentes eletrônicos são sensíveis à eletricidade estática. Para manusear as placas com segurança, você tem de se aterrar antes e durante a instalação. Uma pulseira para aterramento, conectada a uma esteira especial sobre a qual você se ajoelha (a maioria dos computadores requer que você demonstre respeito apropriado!), irá aterrá-lo apropriadamente.

Lembre-se de que você precisa se preocupar com estática quando abre, pela primeira vez, o pacote contendo uma placa de circuito impresso e toda vez que o componente eletrônico for manuseado — não apenas quando da instalação. Tome especial cuidado se a sala onde você recebe sua correspondência (e onde provavelmente você se sentirá tentado a abrir os seus pacotes) tiver carpete; o carpete gera mais eletricidade estática do que um piso frio.

Uma forma de reduzir a estática em pisos acarpetados é adquirir uma garrafa para spray e enchê-la com uma parte de amaciante de tecidos com 10 partes d'água. Borrife-o sobre o carpete (mas não sobre o computador) uma vez por mês para manter baixo os níveis de estática. Esse procedimento também deixa seu escritório com aquele aroma primaveril.

Reassentando placas

Ocasionalmente você mesmo pode resolver um problema de hardware apenas desligando o equipamento, limpando os contatos dos conectores de borda das placas de interface (SCSI, Ethernet etc.), reassentando as placas e ligando o sistema novamente. Se isso funcionar temporariamente, mas o mesmo problema voltar daqui a uma semana ou um mês, o contato elétrico entre o cartão e a placa-mãe provavelmente é inadequado.

Os contatos podem ser limpos com uma solução de limpeza especial e kit de limpeza ou com uma borracha comum. Não use uma borracha velha e dura. Se sua borracha não funcionar bem para apagar traços de lápis numa folha de papel, ela também não funcionará bem em contatos elétricos. Tente manter seus dedos longe dos contatos. Apenas "apague-os" com a borracha (um abrasivo brando), escove os restos de borracha e reinstale a placa. Algumas placas-mãe ainda possuem CIs soquetados, embora isto seja cada vez mais raro. Ao longo do tempo, as conexões nos soquetes podem se deteriorar, em grande parte devido à vibração causada pelas ventoinhas. Você pode pressionar firmemente sobre os chips usando o polegar (depois de ter colocado a pulseira antiestática, logicamente) para comprimi-los e encaixá-los.

25.4 Monitores

Geralmente o monitor é o componente menos confiável dos sistemas computacionais modernos. Muitos monitores possuem ajustes de brilho e de convergência que podem ser acessados somente da placa de circuito impresso. Infelizmente, os monitores normalmente usam cargas internas de dezenas de milhares de volts que podem persistir muito depois de terem sido desconectados da alimentação. Devido ao risco de choque elétrico, recomendamos que você sempre use técnicos qualificados para ajustar os monitores. *Não tente fazer você mesmo esse serviço.*

25.5 Módulos de Memória

O hardware de hoje aceita memória na forma de SIMMs (Single Inline Memory Modules, módulos de memória com pinos de conexão em uma linha), DIMMs (Dual Inline Memory Modules, módulos de memória com pinos de conexão em duas linhas) ou RIMMs (Rambus Inline Memory Modules, módulos de memória com pinos de conexão Rambus) em vez de chips individuais. Esses módulos variam de 32MB a 512MB (1GB em novembro de 2003), todos num pequeno cartão.

Se precisar adicionar memória à estação de trabalho ou servidor, normalmente você mesmo poderá comprá-la de terceiros e instalá-la. Tome cuidado ao adquirir memória de fabricantes de estações de trabalho; seus preços são muitas vezes irreais. Ao adicionar memória, pense grande. O preço da memória está caindo continuamente, mas da mesma forma isto está acontecendo com o número de slots de expansão disponíveis numa placa-mãe típica.

Vale a pena verificar com atenção redobrada a documentação de seu sistema antes de adquirir memória para ter certeza de que você tem uma idéia clara dos tipos de módulos de memória que os seus sistemas aceitarão. Normalmente você pode aumentar o desempenho instalando memória que suporte uma taxa de transferência de barramento mais alta ou recursos especiais como DDR (Double Data Rate, taxa de transferência de dados duplicada). Certifique-se do número de slots de memória que cada sistema tem disponível e se há ou não qualquer restrição no acréscimo de módulos novos. Alguns sistemas requerem que os módulos sejam adicionados aos pares; outros não exigem isto estritamente, mas podem oferecer maior desempenho quando os módulos são usados em pares.

Entenda perfeitamente como os módulos de memória antigos e novos interagirão. Na maioria dos casos, somente os recursos ou velocidades comuns a todos os módulos poderão realmente ser usados. Algumas vezes pode valer a pena eliminar a memória original de um sistema ao fazer uma atualização.

www.snow666.blogspot.com
Snow666

Se você instalar memória por conta própria, tenha em mente as duas regras a seguir:

- As memórias são mais sensíveis à eletricidade estática do que qualquer outra coisa. Certifique-se de estar bem aterrado antes de abrir um saquinho cheio de módulos de memória.
- O conector que é usado para conectar o módulo à placa-mãe varia de máquina para máquina. A maioria normalmente se encaixa facilmente, mas você precisará de uma ferramenta especial para removê-la uma vez instalada. É sempre tentador usar uma caneta esferográfica ou um clipe de papel para soltar as presilhas, mas esta abordagem normalmente acaba danificando o conector.

Os módulos de memória normalmente são candidatos para a tecnologia de limpeza com borracha, descrita anteriormente neste capítulo.

25.6 Manutenção Preventiva

Alguns componentes de hardware possuem filtros que devem ser limpos ou trocados regularmente. Filtros bloqueados impedem o fluxo de ar e pode resultar em sobreaquecimento, uma causa importante de falha em equipamentos. É importante manter abertas e desobstruídas as aberturas de ar em todo o equipamento. Não é raro encontrar livros ou jornais sobre as aberturas de ar de um computador; nestes casos, recomendamos açoitar o infrator repetidamente nas costas até que comece realmente a doer.

Qualquer coisa com partes móveis pode precisar de lubrificação, limpeza e manutenção das correias feitas regularmente. Observe se não há ruídos estranhos provenientes de equipamentos mais antigos e mime-o de acordo.

Em sistemas servidores, as partes que falham com maior freqüência são a ventoinha e o módulo de fonte de alimentação — especialmente em PCs, em que ele normalmente é uma única unidade que pode ser substituída em campo. Verifique periodicamente seus servidores para ter certeza de que suas ventoinhas principais estão rodando de forma rápida e firme. Se não estiverem, normalmente você terá de substituir o bloco de fonte de alimentação inteiro — caso contrário, você corre o risco de sobreaquecer o seu equipamento. *Não* tente lubrificar a própria ventoinha; este procedimento poderia adiar a quebra inevitável, mas também poderia acelerar o problema ou provocar danos a outros componentes.

Muitos gabinetes de PC fornecem um local conveniente para montagem de uma segunda ventoinha (e conexões elétricas para alimentá-lo). Se o ruído não for problema, é sempre recomendável instalar uma segunda ventoinha. Além de diminuir a temperatura de operação dos componentes, a ventoinha extra atua como um backup no caso da ventoinha principal falhar. Ventoinhas extras são baratas; tenha em mãos algumas delas como peças de reposição.

Um computador num ambiente empoeirado queimará componentes com muito maior freqüência do que um cujo ambiente é relativamente limpo. A poeira bloqueia filtros, resseca partes lubrificadas, bloqueia partes móveis (ventoinhas) e recobre os componentes com uma camada "isolante" de poeira que reduz a capacidade destes componentes de dissipar calor. Todos esses efeitos tendem a aumentar as temperaturas operacionais. Verifique e elimine regularmente o pó dos filtros. Talvez você também precise de vez em quando dar uma bela limpeza nas partes internas de seu sistema quando estiverem em ambientes desfavoráveis. (Qualquer ambiente dotado de carpete é forte candidato a ser ruim.)

Aspirar é a melhor maneira de remover poeira, porém mantenha o motor do aspirador a pelo menos um metro e meio dos componentes e discos para minimizar a interferência magnética. O ambiente onde se encontram suas máquinas deve ser aspirado regularmente, mas garanta que tal tarefa seja executada por pessoas que foram treinadas a respeitar distâncias apropriadas a fim de não danificarem os equipamentos (o pessoal de limpeza normalmente não é candidato aceitável para esta tarefa).

As unidades de fita normalmente também requerem limpeza regular. A maioria das unidades de fita é limpa usando-se uma fita de limpeza especial.

25.7 Ambiente

Da mesma forma que os seres humanos, os computadores trabalham melhor e por mais tempo se estiverem felizes em seus ambientes de trabalho. Embora não liguem muito para ter uma janela com vista, eles realmente prestam atenção a outros aspectos de seus lares.

Temperatura

A temperatura ideal para operação de equipamentos de informática fica na faixa dos 17° a 20°C, com cerca de 45% de umidade. Infelizmente, essa temperatura não coincide com a temperatura de operação ideal dos usuários de computadores. Temperaturas ambientes acima de 27°C na sala dos equipamentos implicam cerca de 49°C dentro

Snow666

das máquinas. Chips comerciais apresentam um intervalo operacional de até cerca de 49°C, em cujo ponto eles pararão de funcionar; acima de 71°C, eles quebrarão.

Umidade

A umidade ideal para a maior parte do hardware de informática está na faixa de 40% a 60%. Se a umidade for muito baixa, a eletricidade estática se torna um problema. Se ela for muito alta, pode se formar condensação sobre as placas, provocando curtos e oxidação.

Refrigeração do escritório

Nos dias de hoje, muitos computadores residem nas salas das pessoas e têm de sobreviver sob ar-condicionado (normalmente desligado à noite e nos fins de semana) e têm de suplantar uma boa dose de papéis e de livros sobre suas aberturas para ventilação. Ao colocar um computador numa sala, tenha em mente que ele roubará ar-condicionado destinado aos seres humanos. Se você se encontrar numa função na qual poderá influenciar na capacidade de refrigeração, uma regra prática é que cada ser humano na sala produz 300 BTUH de calor, ao passo que um PC produz em média 1.100 BTUH. Não deixe os técnicos esquecerem de levar em conta a carga solar proveniente de janelas que recebem diretamente luz solar.

Refrigeração de salas de máquinas

Se tiver "sorte" bastante para deslocar sua horda de servidores numa daquelas extravagantes salas de piso falso construídas na década de 80 que tinha capacidade para refrigerar todo o seu equipamento *e* o Estado de Oklahoma juntos, então sua maior preocupação provavelmente será encontrar algum material instrutivo em manutenção de sistemas de refrigeração. Para o restante de nós, dimensionar corretamente um sistema de refrigeração é o que faz diferença a longo prazo. Uma sala de máquinas bem refrigerada é uma sala de máquinas exuberante.

Descobrimos que é uma boa idéia verificar com atenção redobrada as estimativas de carga de refrigeração fornecida pelo pessoal da área de sistemas de refrigeração, especialmente ao instalar um sistema para uma sala de máquinas. Certamente você precisará de um especialista na área para ajudá-lo a calcular a carga de refrigeração que tetos, paredes e janelas (não se esqueça da carga solar) contribuem para o ambiente. Os especialistas em sistemas de refrigeração normalmente têm muita experiência com esses componentes e devem estar aptos a lhe dar uma estimativa precisa. A parte que você precisa verificar é a carga de calor interna da sala de máquinas.

Você precisará determinar as parcelas de contribuição de carga de calor para os seguintes componentes:

- Teto, paredes e janelas (consulte um especialista em sistemas de refrigeração para esta estimativa)
- Equipamentos eletrônicos
- Acessórios de iluminação
- Operadores (pessoas)

Equipamento eletrônico

Você pode estimar a carga de calor produzida pelos seus servidores (e outros dispositivos eletrônicos) determinando seu consumo de potência. Medição direta de consumo de potência é de longe o melhor método para obter essas informações. Seu amigo eletricista mais próximo poderá ajudá-lo nisso. De outra forma, a maioria dos equipamentos apresentam etiquetas mostrando o seu consumo de potência máximo em watts, embora o consumo típico tenda a ser menor do que o máximo. Você poderá converter esses números em unidades de calor padrão, BTUH, multiplicando por 3,412 BTUH/watt. Por exemplo, se quiséssemos construir uma sala de máquinas que fosse abrigar 25 servidores classificados em 450 watts cada, o cálculo ficaria assim:

$$\left(25\ \text{servidores}\right)\left(\frac{450\ \text{watts}}{\text{servidores}}\right)\left(\frac{3{,}412\ \text{BTUH}}{\text{watts}}\right)=38.385\ \text{BTUH}$$

Acessórios de iluminação

Como acontece com os equipamentos eletrônicos, você pode estimar a carga calorífica de acessórios de iluminação baseado no consumo de potência. Os acessórios de iluminação típicos para escritório contêm quatro lâmpadas fluorescentes de 40 watts cada. Se a sua sala tivesse seis desses acessórios, o cálculo ficaria assim:

$$\left(6\ \text{acessórios de iluminação}\right)\left(\frac{160\ \text{watts}}{\text{acessórios}}\right)\left(\frac{3{,}412\ \text{BTUH}}{\text{watts}}\right)=3.276\ \text{BTUH}$$

www.snow666.blogspot.com
Snow666

Operadores
De vez em quando, seres humanos precisarão entrar na sala das máquinas para fazer alguma coisa. Calcule 300 BTUH para cada ocupante. Se quiser estimar quatro pessoas na sala das máquinas ao mesmo tempo:

$$\left(4 \text{ pessoas}\right)\left(\frac{300 \text{ BTUH}}{\text{pessoas}}\right) = 1.200 \text{ BTUH}$$

Carga calorífica total
Assim que tiver calculado a carga calorífica para cada componente, adicione-a para determinar a carga calorífica total. Para nosso exemplo, supomos que nosso especialista em refrigeração estimou uma carga calorífica total proveniente de teto, paredes e janelas de 20,000 BTUH.

20,000	BTUH para teto, paredes e janelas
38,385	BTUH para servidores e outros equipamentos eletrônicos
3,276	BTUH para acessórios de iluminação
1,200	BTUH para operadores
62,861	BTUH total

A capacidade de um sistema de refrigeração é tipicamente expressa em toneladas. Podemos converter BTUH para toneladas dividindo por 12.000 BTUH/ton. Você também deve permitir um fator de pelo menos 50% para levar em conta erros e crescimento futuro:

$$\left(62.681 \text{ BTUH}\right)\left(\frac{1 \text{ ton}}{12.000 \text{ BTUH}}\right)\left(1{,}5\right) = 7.86 \text{ ton de refrigeração necessárias}$$

Veja em que grau sua estimativa coincide com aquela fornecida pelo pessoal especializado em refrigeração.

Monitoramento de temperatura

Se estiver suportando um ambiente computacional crítico, é uma boa idéia monitorar a temperatura (e outros fatores ambientais como ruído e potência) na sala de máquinas mesmo quando você não estiver lá. Poderá ser muito frustrante chegar na segunda-feira pela manhã e encontrar um monte de plástico derretido no chão da sala de máquinas. Felizmente, monitores de salas de máquinas automatizados podem observar as coisas enquanto você está fora. Usamos e recomendamos a família de produtos Phonetics Sensaphone. Esses computadores baratos monitoram variáveis de ambiente como temperatura, ruído e potência e eles telefonam para você (ou no seu pager) quando for detectado algum problema. Para contactar a Phonetics em Aston, PA, Estados Unidos, vá ao site Web em www.sensaphone.com.

25.8 POTÊNCIA

Os equipamentos de informática gostariam de receber alimentação regular, estável e limpa. Numa sala de máquinas, isso significa um estabilizador de potência, um caro equipamento que filtra picos e pode ser ajustado para fornecer níveis de tensão e fases corretos. Em escritórios, protetores contra surtos colocados entre as máquinas e as tomadas ajudarão a isolar os equipamentos dos picos de tensão.

Servidores e equipamentos de infra-estrutura de rede poderiam ser colocados num no-break. No-breaks de qualidade possuem uma interface RS-232, Ethernet ou USB que pode ser conectada à máquina a qual eles fornecem alimentação. Essa conexão permite que o no-break alerte o computador de que a alimentação falhou e que ele deve se desligar sozinho e de forma ordenada antes das baterias acabarem.

Consulte a seção 2.5 para obter maiores informações sobre procedimentos de desligamento.

Um estudo estimou que 13% da energia elétrica consumida nos Estados Unidos é usada para alimentar computadores. Tradicionalmente, computadores UNIX se baseavam em hardware e software que esperavam que a alimentação estivesse disponível 24 horas por dia. Nos dias de hoje, somente servidores e dispositivos de rede realmente precisam estar ativos todo o tempo. As máquinas desktop podem ser desligadas à noite se há um modo fácil para os seus usuários fazerem isto (e se você confiar nos seus usuários para fazê-lo corretamente).

No mínimo, solicite a seus usuários para desligar os monitores e impressoras laser quando forem para casa, pois estes são os maiores consumidores de energia. Ao comprar equipamento novo, procure a certificação Energy Star. Isso significa que um item atende às diretrizes do EPA para operação eficiente em termos de energia. Por exemplo, os monitores Energy Star têm de ser capazes de desligarem automaticamente suas telas após um determinado período de inatividade.

www.snow666.blogspot.com
Snow666

Controle remoto de potência

Ocasionalmente você pode se deparar com a situação na qual tenha de ligar/desligar um servidor devido a uma falha de hardware ou de kernel. Ou quem sabe você tenha servidores não Linux em sua sala de máquinas mais suscetíveis a esse tipo de problema. Em ambos os casos talvez seja interessante instalar um sistema que lhe permita efetuar o ciclo liga/desliga[1] em servidores problemáticos via controle remoto.

Uma solução barata e popular é o sistema de controle de potência X-10, que inclui um computador que responderá a uma linha telefônica e efetuará o ciclo liga/desliga das máquinas baseado em códigos de tons. A linha de produtos básica X-10 é descrita e vendida on-line no site www.x10.com. Caso esteja realmente interessado nesse tipo de abordagem, certifique-se de verificar o que há disponível em www.smarthome.com, que se concentra basicamente em automação doméstica, mas inclui equipamentos compatíveis com X10 de maior qualidade.

Uma solução mais sofisticada (e cara) é fabricada pela APC (American Power Conversion). O seu produto MasterSwitch é similar a uma régua de alimentação, exceto pelo fato de poder ser controlado por um navegador Web através de sua porta Ethernet embutida. Para acessar a APC ligue para (00xx1) (401) 789-0204 ou no seu site Web em www.apcc.com.

25.9 RACKS

Para grande desalento daquele pessoal de Wall Street que uma vez já viu pisos falsos como um símbolo de status corporativo comparável a ter uma Lamborghini, os dias das verdadeiras salas de máquinas com piso falso acabaram. Alguma vez você já tentou puxar um cabo que passa por baixo do piso de um desses mastodontes? Nossa experiência diz que embora pareça bonito através daqueles compartimentos de vidro, "piso falso" é sinônimo para "ninho de ratos". Caso tenha *obrigatoriamente* de instalar um piso falso, use-o para ocultar as fontes de alimentação e *nada mais*.

Se seu objetivo é operar seu equipamento de uma forma profissional, uma sala de máquinas dedicada para máquinas da categoria servidor é essencial. Uma sala de servidores não somente fornece um ambiente aconchegante de temperatura controlada para suas máquinas, como também atende suas necessidades de segurança física.

Numa sala de máquinas dedicada, armazenar equipamentos em racks (ao contrário de, digamos, dispô-los em mesas ou no chão) é a única opção profissional e de fácil manutenção. Hoje, os melhores esquemas de armazenamento usam racks que são interconectados num sistema de trilhas suspensas para passagem de cabos. Essa abordagem fornece aquela sensação irresistível de alta tecnologia sem sacrificar a organização ou a manutenção.

O melhor desse tipo de sistema é fabricado pela Chatsworth Products (Chatsworth, CA, (818) 882-8595). Usando racks de telecomunicação trilha única padrão 19" você pode construir abrigos para servidores montados em prateleiras e em servidores montados em racks. Dois racks de telecomunicação de 19" com suas partes traseiras encostadas umas nas outras compõem um rack "tradicional" de aspecto de alta tecnologia (para casos nos quais você precise conectar racks tanto pela frente como pela traseira do equipamento). A Chatsworth fornece os racks, canaletes para cabos e outros bibelôs para gerenciamento de cabos, bem como o hardware necessário para montá-los em seu prédio. Já que os cabos repousam em trilhas visíveis, eles são fáceis de serem acompanhados e você naturalmente será motivado a mantê-los organizados.

25.10 FERRAMENTAS

Um administrador de sistemas bem equipado é um administrador de sistemas eficiente. Ter uma caixa de ferramentas dedicada é um fator importante para minimizar o tempo desligado no caso de uma emergência. A Tabela 25.1 lista alguns itens que você provavelmente deve manter em sua caixa de ferramentas ou pelo menos a fácil alcance.

Tabela 25.1 A caixa de ferramentas de um administrador de sistemas.

Ferramentas genéricas	
Chaves Phillips: #0, #1 e #2	Pinças
Chaves de fenda: 1/8", 3/16" e 5/16"	Tesoura
Faca de eletricista ou canivete suíço	Kit de chaves para aperto de soquetes
Alicates, comum e de bico	Lanterna pequena ou tipo caneta
Chaves de fenda de joalheiro	Kit de chaves sextavadas
Martelo de bola, 120g	Kit de chaves de torção

1. O ciclo liga/desliga é o processo de desligar uma máquina, aguardar 30 a 60 segundos para os capacitores descarregarem e ligar a máquina novamente.

Snow666

Tabela 25.1 A caixa de ferramentas de um administrador de sistemas. (*continuação*)

Itens especiais relacionados com equipamentos de informática	
Descascador de fios (com um cortador de fios integrado)	Analisador de redes Fluke
Presilhas para cabos (e seus primos em velcro)	Extrator de Cis
Cabos de transposição Categoria RJ-45 sobressalentes	Plissador de terminações RJ-45
Conectores RJ-45 sobressalentes (para fios maciço e trançado)	Terminadores SCSI
Multímetro digital	Dispositivo para desligamento
Pulseira antiestática	
Variedades	
Lista de contatos para emergências de manutenção[a]	Q-Tips
Números de telefones residenciais e de pager do pessoal de suporte por chamada	Fita isolante
Kit de primeiros socorros	Espelhinho de dentista
Seis pacotes de cerveja de qualidade	Telefone celular

a. E números dos contratos de manutenção se aplicável.

25.11 EXERCÍCIOS

E25.1 Por que você iria querer montar seus computadores num rack?

★ **E25.2** Fatores ambientais afetam tanto pessoas quanto máquinas. Amplie os fatores listados neste livro com alguns fatores próprios (por exemplo, poeira, ruído, luminosidade, entupimentos etc). Escolha quatro fatores e avalie a adequação de sua instalação para as pessoas e as máquinas.

★ **E25.3** Uma estação de trabalho puxa 0,8 A, e seu monitor puxa 0,7 A em 120V.

a) Qual a potência em watts consumida por este sistema? (Dica: P = UI)

b) Com o custo da eletricidade por volta dos US$ 0,10/kWh, quanto custa deixar este sistema ligado o ano inteiro?

c) Qual a quantia que pode ser poupada anualmente ao desligar-se o monitor por uma média de 16 horas por dia (seja manualmente ou usando-se recursos Energy Star como o de Sinalização para Gerenciamento de Energia de Monitores)?

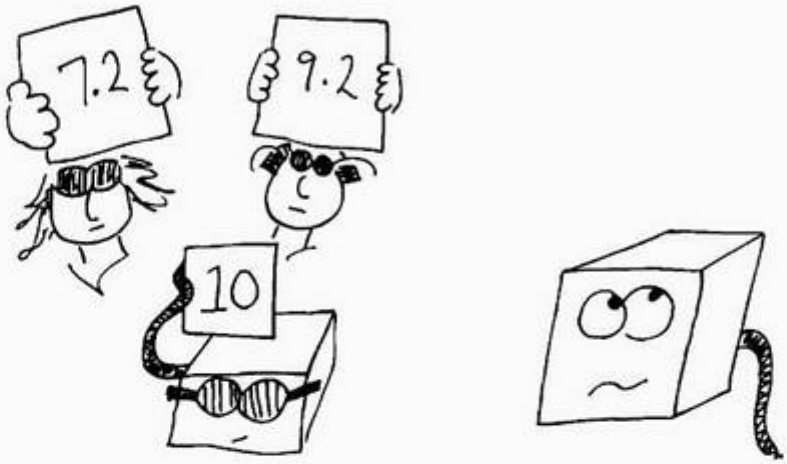
d) Qual o custo anual para refrigerar este sistema? (Exponha suas suposições referentes aos custos de refrigeração e mostre os seus cálculos.)

★★ **E25.4** Projete um novo laboratório de informática para a sua instalação. Exponha suas suposições referentes a espaço, números de máquinas e tipo e carga de potência de cada máquina.

A seguir, calcule as necessidades de potência e de refrigeração para o laboratório. Inclua tanto os servidores quanto as estações de trabalho. Inclua também o layout da sala, a iluminação e a carga calorífica referentes às pessoas que ocuparão a sala.

www.snow666.blogspot.com
Snow666

26 Análise de Desempenho



O desempenho é uma das características mais visíveis de qualquer sistema e freqüentemente se encontra no alto da lista de reclamações dos usuários. Muitos usuários estão convencidos de que seus computadores poderiam rodar duas vezes mais rápido bastando, para tanto, que o administrador soubesse como ajustar apropriadamente o sistema para liberar o seu vasto e intacto potencial. Na realidade, isso quase nunca é verdade.

Uma fantasia comum envolve o ajuste fino das variáveis de kernel que controlam o sistema de paginação e os reservatórios de buffers. Em tempos passados, havia situações nas quais esse ajuste era prudente e necessário. Nos dias de hoje, normalmente é uma má idéia. O resultado mais provável é que você irá *reduzir* o desempenho global de seu sistema e nem mesmo estará ciente do que você fez, durante todo este tempo se parabenizando por ser um administrador tão astuto.

Os kernels das principais distribuições são pré-ajustados para atingir desempenho razoável (embora admitidamente, não ótimo) sob uma série de condições de carga. Se você tentar otimizar o sistema tomando como base uma medida de desempenho em particular, muito provavelmente estará distorcendo o comportamento do sistema em relação a outras métricas de desempenho e condições de carga. Parece fácil obter resultados, mas os ganhos normalmente são ilusórios.

Particularmente, tome tudo o que você lê na Web com uma colher de sopa de sal, isto é, com muita cautela. Na área de desempenho de sistemas, você verá argumentos superficialmente convincentes sobre todos os tipos de tópicos. Entretanto, a maioria dos proponentes dessas teorias não possui conhecimento, disciplina e tempo necessários para elaborar experimentos válidos. Suporte popular significa muito pouco; para cada proposta estúpida, você pode esperar ver um coro grego: "Eu multipliquei o tamanho do meu cache de buffers por um fator de dez vezes, exatamente como disse o Joe e o meu sistema está MUITO, MAS MUITO mais rápido!!!". Está bem.

O desempenho dos sistemas não está totalmente fora do seu controle. Só que o caminho para um bom desempenho não é pavimentado com correções mágicas e correções de kernel românticas. As regras básicas são as seguintes:

- Coletar e revisar informações *históricas* sobre o seu sistema. Se ele estava funcionando bem há uma semana, um exame dos aspectos do sistema que recentemente foram alterados provavelmente o levarão a uma prova irrefutável do crime. Preserve tendências e linhas de referência regularmente em sua manga para usá-la numa emergência.
- Não sobrecarregue seus sistemas ou rede. O Linux dá a cada processo uma ilusão de recursos infinitos. Mas tão logo 100% dos recursos do sistema estiverem em uso, o Linux terá de trabalhar muito para manter esta ilusão, atrasando processos e normalmente consumindo uma parte razoável dos próprios recursos.

Snow666

Este capítulo se concentra no desempenho dos sistemas que são usados como servidores. Sistemas desktop normalmente não experimentam os mesmos tipos de problemas de desempenho pelos quais os servidores passam, e a resposta à questão de como aumentar o desempenho numa máquina desktop é quase sempre "Atualize o hardware". Os usuários gostam dessa resposta, pois isso significa que eles ganham sistemas novinhos em folha com maior freqüência.

26.1 O Que Você Pode Fazer para Melhorar o Desempenho

Eis algumas coisas específicas que podem ser feitas para melhorar o desempenho:

- Certificar-se de que o sistema tem memória suficiente. Como veremos a seguir, o tamanho da memória tem uma influência fundamental sobre o desempenho. A memória é tão barata nos dias de hoje que normalmente se pode pagar para ter carga máxima de memória em todas as máquinas sensíveis a questões de desempenho.
- Verificar com atenção redobrada a configuração do sistema e de cada aplicativo. Muitos aplicativos podem ser ajustados de diversas maneiras, levando a ganhos incríveis de desempenho (por exemplo, distribuindo dados entre os discos, não realizar pesquisas DNS em tempo de execução ou executando um número maior de instâncias de um servidor popular).
- Corrigir problemas de utilização, tanto aqueles causados pelos usuários (muitas tarefas sendo executadas ao mesmo tempo, práticas de programação ineficiente, tarefas executadas com prioridade excessiva e tarefas volumosas executadas em momentos inapropriados do dia) quanto aqueles provocados pelo sistema (cotas, contabilidade de CPU, daemons indesejados).
- Para casos nos quais estiver usando o Linux como um servidor Web ou como algum outro tipo de servidor de aplicativos de rede, talvez seja interessante distribuir o tráfego entre uma série de sistemas com um aplicativo de balanceamento de carga comercial como o Content Services Switch da Cisco (www.cisco.com) ou o Alteon Web Switch da Nortel (www.nortelnetworks.com).[1] Esses aplicativos fazem com que vários servidores físicos pareçam ser um único servidor lógico para o mundo externo. Eles balanceiam a carga de acordo com um dos vários algoritmos selecionáveis pelo usuário como "servidor mais responsivo" ou "round robin". Esses balanceadores de carga também fornecem redundância prática caso um servidor tenha de ser desligado. Eles são realmente muito necessários se a sua instalação tiver de lidar com picos de tráfego inesperados.
- Organizar os discos rígidos e sistemas de arquivos de maneira que a carga seja balanceada igualmente, maximizando o desempenho de I/O. Para aplicativos específicos como bancos de dados, você poderá usar uma tecnologia multidisco engenhosa como RAID para otimizar transferências de dados. Consulte seu fornecedor de bancos de dados para recomendações.

  É importante notar que os diferentes tipos de aplicativos e bancos de dados respondem de modo diferente quando distribuídos entre vários discos. O RAID vem em várias formas e você terá de se empenhar para determinar qual forma (se realmente existir alguma) é apropriada para o seu aplicativo particular.
- Monitorar a rede para certificar-se de que ela não esteja saturada com tráfego e que a taxa de erros seja baixa. Uma enorme quantidade de informações sobre rede se encontra disponível por meio do comando **netstat**, descrito na seção 20.4. Veja também o Capítulo 20, *Gerenciamento e Depuração de Redes*.
- Configurar o kernel para eliminar drivers e opções indesejadas e usar tabelas de tamanho apropriado. Esses tópicos são cobertos no Capítulo 12, *Drivers e o Kernel*.
- Identificar situações nas quais o sistema esteja fundamentalmente inadequado para satisfazer às demandas solicitadas.

Esses passos são listados grosseiramente em ordem de eficiência. Adicionar memória e balancear tráfego entre os servidores pode fazer uma enorme diferença em termos de desempenho. Pode ser que haja alguma melhoria organizando-se corretamente os discos do sistema e corrigindo problemas de rede. Pode ser que os demais fatores não façam nenhuma diferença.

A otimização de algoritmos e estruturas de dados de software quase sempre leva a ganhos de desempenho significativos. Mas a menos que você tenha uma base substancial de software local, esse nível de projeto normalmente fica fora de controle.

26.2 Fatores Que Afetam o Desempenho

O desempenho percebido é determinado pela eficiência com a qual os recursos do sistema são alocados e compartilhados. A definição exata de um "recurso" é um tanto vaga. Ela pode incluir itens como contextos em cache no chip

1. Uma alternativa gratuita, porém um tanto menos estável, é o software Linux Virtual Server disponível em linuxvirtualserver.org.

Snow666

da CPU e entradas na tabela de endereços do controlador de memória. Entretanto, para uma primeira abordagem, somente os quatro recursos a seguir têm grande efeito sobre o desempenho:

- Tempo de CPU
- Memória
- I/O de disco rígido
- I/O de rede

Todos os processos consomem parte dos recursos do sistema. Se ainda forem deixados recursos depois de processos ativos terem pego aquilo que desejavam, o desempenho do sistema será quase tão bom quanto aquele que normalmente deveria acontecer.

Se não houver recursos suficientes para usar, os processos têm de ser executados em turnos. Um processo que não tem acesso imediato a recursos de que ele precisa tem de aguardar sem fazer nada. A quantidade de tempo gasta aguardando é uma das medidas básicas de degradação de desempenho.

O tempo de CPU é um dos recursos mais fáceis de serem medidos. Uma quantidade constante de poder de processamento está sempre disponível. Teoricamente, essa quantidade é 100% dos ciclos de CPU, mas o overhead e diversas ineficiências fazem que o número prático seja mais provavelmente 95%. Um processo que esteja usando mais do que 90% dos recursos de CPU está inteiramente vinculado à CPU e consumindo grande parte do poder de processamento disponível no sistema.

Muitas pessoas supõem que a velocidade da CPU é o fator mais importante afetando o desempenho global de um sistema. Dadas as quantidades infinitas de todos os demais recursos ou de certos tipos de aplicativos (por exemplo, simulações numéricas), uma CPU mais rápida *fará* uma diferença dramática. Mas no mundo prático, a velocidade de CPU relativamente não é importante.

O gargalo mais comum em termos de desempenho em sistemas Linux é, na verdade, largura de banda em disco. Pelo fato de os discos rígidos serem sistemas mecânicos, leva muitos milissegundos para se localizar um bloco em disco, recuperar o seu conteúdo e acordar o processo que está aguardando por ele. Atrasos dessa magnitude ofuscam cada uma das demais fontes de degradação de desempenho. Cada acesso a disco dispara milhões de instruções de CPU.

Pelo fato de o Linux fornecer memória virtual, a largura de banda em disco e a memória estão diretamente relacionadas. Num sistema carregado com uma quantidade limitada de RAM, normalmente temos de gravar uma página em disco para obter uma página atualizada de memória virtual. Infelizmente, isso significa que usar memória é normalmente tão caro quanto usar o disco. A paginação provocada por software pesado é o inimigo número 1 na maioria das estações de trabalho.

A largura de banda de rede lembra a largura de banda em disco em diversos aspectos, devido às latências envolvidas. Entretanto, as redes são atípicas no aspecto que elas envolvem comunidades inteiras em vez de computadores individuais. Elas também são suscetíveis a problemas de hardware e a servidores sobrecarregados.

26.3 Verificação de Desempenho do Sistema

A maioria das ferramentas para análise de desempenho nos informa sobre o que está acontecendo em um determinado ponto no tempo. Entretanto, o número e o caráter das cargas provavelmente mudarão ao longo do dia. Certifique-se de coletar um grupo representativo de dados antes de tomar uma atitude. As melhores informações sobre desempenho de sistema normalmente se tornam claras somente após um longo período (um mês ou mais) de coleta de dados. É particularmente importante coletar dados durante períodos de pico de uso. As limitações de recursos e de configurações incorretas de sistema normalmente são visíveis somente quando a máquina se encontra sob uma carga pesada.

Analisando o emprego de CPU

Provavelmente você vai querer coletar três tipos de dados de CPU: utilização global, médias de cargas e consumo de CPU por processo. A utilização global pode ajudar a identificar sistemas nos quais a velocidade da CPU em si é o gargalo. As médias de carga lhes dão uma idéia do desempenho do sistema como um todo. Os dados de consumo da CPU por processo podem identificar processos específicos que estão devorando recursos.

Para obter informações sintetizadas, utilize o comando **vmstat**. **vmstat** usa dois argumentos: o número de segundos para monitorar o sistema para cada linha de saída e o número de relatórios a fornecer. Se você não especificar o número de relatórios, **vmstat** será executado até você pressionar <Control-C>. A primeira linha de dados retornada por **vmstat** fornece médias desde o momento em que o sistema foi inicializado. As linhas a seguir são medidas dentro do período de amostragem anterior, cujo padrão é de cinco segundos. Por exemplo:

Snow666


```
% vmstat 5 5
procs        memory         swap io    system   cpu
r b w swpd free buff cache si so bi bo in  cs us sy id
2 0 0 5288 1896 9352 17868 0   0  0  1  27 12  2  0 98
0 0 0 5392 1360 8300 18804 0   0  0  0  21 80  1  1 98
0 0 1 5812 1032 9172 18668 0  18  9 31 219 31 79 20  1
1 0 0 5304  900 3604 17660 0   0  0 18  96 16 35 59  7
2 0 0 5268 1440 4148 20056 0   0  3 11 314 48 52 48  0
```

Tempo de usuário, de sistema e ocioso são mostrados nas colunas `us`, `sy` e `id` bem à direita. Números de CPU grandes na variável de tempo de usuário geralmente indicam processamento, e números de sistema elevados indicam que os processos estão fazendo um grande número de chamadas de sistema ou executando I/O. Uma regra prática que tem nos servido ao longo dos anos é a de que o sistema deve gastar cerca de 50% de seu tempo não ocioso em espaço de usuário e 50% em espaço de sistema; a porcentagem de tempo ocioso global não deve ser nula.

A coluna `cs` ilustra trocas de contexto por intervalo, o número de vezes que o kernel mudou enquanto o processo estava sendo executado. O número de interrupções por intervalo (normalmente gerado por dispositivos de hardware ou componentes do kernel) é ilustrado na coluna `in`. Valores `cs` ou `in` extremamente elevados normalmente indicam um dispositivo de hardware se comportando mal ou mal configurado. As demais colunas são úteis para análise de memória e de disco, que discutiremos posteriormente.

Médias a longo prazo das estatísticas de CPU permitem que você determine se fundamentalmente há poder de processamento suficiente para satisfazer a demanda. Se a CPU normalmente gasta parte de seu tempo no estado ocioso, há ciclos para reserva. Atualizar para uma CPU mais rápida não irá aumentar muito o desempenho global do sistema, embora ele possa acelerar operações individuais.

Como pode ser observado nesse exemplo, a CPU geralmente fica alternando entre uso pleno e ociosidade completa. Portanto, é importante observar esses números como uma média ao longo do tempo. Quanto menor for o intervalo de monitoramento, menos consistentes serão os resultados.

Em máquinas multiprocessadas, a maioria das ferramentas Linux apresenta uma média de estatística de processador ao longo de todos os processadores. O comando **mpstat** gera saída parecida com **vmstat** para cada processador. O flag **-P** permite que você especifique um determinado processador sobre o qual será gerado o relatório. **mpstat** é útil para depuração de software que suporte SMP (Symmetric MultiProcessing, multiprocessamento simétrico) — também é esclarecedor ver se o sistema está usando processadores múltiplos de maneira eficiente ou ineficiente.

Numa estação de trabalho com somente um usuário, a CPU geralmente gasta 99% de seu tempo em estado ocioso. Portanto, quando você for rolar uma das janelas em seu display bitmap, a CPU é derrubada por um curto período. Nessa situação, as informações sobre emprego médio de CPU a longo prazo não são significativas.

O segundo tipo de informações estatísticas sobre CPU útil para caracterizar a carga no sistema é a "média de carga", o número médio de processos que podem ser executados. Em geral, a média de carga inclui processos que estão aguardando por I/O de disco ou de rede, de modo que não se trata de uma medida pura de emprego de CPU. Entretanto, ela dá efetivamente uma boa idéia do número de fatias em que a CPU está sendo dividida. A média de carga é obtida com o comando **uptime**:

```
% uptime
11:10am up 34 days, 18:42, 5 users, load average: 0.95, 0.38, 0.31
```

São dados três valores, correspondentes a médias de 5, 10 e 15 minutos. Em geral, quanto maior for a média de carga, mais importante se torna o desempenho agregado do sistema. Se houver somente um processo executável, este processo normalmente estará vinculado por um único recurso (comumente largura de banda de disco ou CPU). A demanda de pico para esse recurso em particular se torna o fator determinante no desempenho.

Quando um número maior de processos estiver compartilhando o sistema, as cargas podem ou não ser igualmente distribuídas. Se os processos no sistema consumirem uma mistura de CPU, de disco e de memória, o desempenho do sistema será menos provavelmente dominado por restrições num único recurso. Nessa situação, torna-se mais importante observar as medidas médias de consumo como a utilização total de CPU.

Consulte a seção 5.3 para obter maiores informações sobre prioridades.

Os sistemas Linux estão atarefados com uma média de carga 3 e não se dão bem com médias de carga acima de 6 aproximadamente. Uma média de carga dessa magnitude é uma dica de que você deve começar a procurar maneiras de distribuir artificialmente a carga, como solicitar aos usuários para executar processos longos à noite ou usar **nice** para configurar prioridades de processo.

www.snow666.blogspot.com
Snow666

A média de carga do sistema é uma excelente métrica para se acompanhar como parte de uma linha de referência do sistema. Se você souber a média de carga de seu sistema num dia normal e esta estiver na mesma faixa num dia ruim, isto é indicativo de que você deve procurar algum outro fator como causa dos problemas de desempenho (como, por exemplo, a rede). Uma média de carga acima do padrão esperado sugere que você deve verificar os processos em execução no próprio sistema Linux.

Uma outra maneira de verificar o emprego de CPU é executar o comando **ps** com os argumentos **aux**, que lhe mostram quanto de CPU cada processo está usando. Num sistema atarefado, pelo menos 70% da CPU normalmente será consumida por apenas um ou dois processos. (Lembre-se de que o próprio **ps** consome um pouco de recursos da CPU.) Adiar a execução de grandes consumidores de recursos de CPU ou reduzir suas prioridades fará com que a CPU se torne mais disponível para outros processos.

*Consulte a seção 5.5 para obter maiores informações sobre **top**.*

Uma excelente alternativa para **ps** é um programa chamado **top**. **top** apresenta praticamente as mesmas informações que **ps**, mas num formato "vivo" que lhe permite observar o estado da mudança do sistema ao longo do tempo.[2]

Como o Linux gerencia a memória

Antes de mergulharmos nos detalhes do gerenciamento de memória no Linux, vale a pena mencionar mais uma vez que o kernel Linux evolui rapidamente, mais rapidamente do que qualquer outro sistema operacional. Na época em que este livro era escrito, o sistema de gerenciamento de memória no Linux estava num período de fluxo especialmente rápido, mesmo para os padrões do Linux. O sistema de gerenciamento de memória inteiro foi substituído entre as versões 2.2.x e 2.4.0 do kernel e ele foi novamente substituído por código novo no kernel versão 2.4.10. Vimos mais alterações nos kernels 2.4.x mais recentes, e o kernel versão 2.5 supostamente usará um sistema de gerenciamento de memória completamente novo. Esta seção fornece uma descrição dos conceitos atuais de gerenciamento de memória Linux que devem ser úteis na compreensão de seu kernel Linux independentemente dos detalhes específicos de cada versão.

Assim como outros sistemas UNIX, o Linux gerencia memória em unidades chamadas páginas. O tamanho atual de uma página de memória no PC é de 4K. O kernel do Linux aloca páginas virtuais para processos à medida que eles forem solicitando memória. Cada página virtual é associada a armazenamento real, seja RAM ou espaço de swap em disco. O Linux usa uma "tabela de páginas" para controlar a associação entre estas páginas virtuais e páginas reais de memória. O Linux pode alocar efetivamente tanta memória quanto os processos solicitarem ampliando a RAM real com espaço de swap. Já que os processos esperam que suas páginas virtuais se associem à memória real, o Linux está constantemente atarefado trocando de posição entre páginas RAM e de swap. Essa atividade é conhecida como paginação.

O Linux tenta gerenciar a memória do sistema para que as páginas que foram acessadas recentemente são mantidas em memória e páginas menos ativas sejam colocadas em disco. Esse esquema é conhecido como sistema LRU, já que as páginas usadas menos recentemente são aquelas excluídas. Seria muito ineficiente para o kernel realmente controlar todas as referências de memória, de modo que o Linux usa um cache de páginas para decidir que páginas excluir. Esse sistema é muito mais barato do que um sistema LRU verdadeiro, mas produz resultados similares.

O Linux mantém um controle de uma "idade" para cada página de memória virtual. Toda vez que o Linux examina uma página e descobre que ela foi recentemente referida, ele incrementa a idade da página. (O termo "idade" é um tanto incorreto, pois o valor realmente indica a freqüência e quão recente foi o acesso. Quanto mais alta for a idade, mais recente será a página.) Enquanto isso, o Linux executa o processo **kswapd**, que diminui regularmente as idades de páginas não referidas cortando-as ao meio.

O kernel mantém várias listas de páginas de memória. Páginas com uma idade maior do que zero são marcadas como "ativas" e estão contidas no cache de páginas "de lista ativa". Se a idade de uma página atingir zero, **kswapd** intercambiará a página para a "lista inativa". **kswapd** remove páginas inativas da tabela de páginas e as considera elegíveis para serem paginadas para disco. Embora páginas inativas não sejam mais imediatamente acessíveis através da tabela de páginas, o kernel irá recuperá-las da memória ou disco e as colocará de volta na tabela de páginas sob demanda.

Quando os níveis de memória estiverem baixos, o kernel tentará descobrir quais páginas na lista inativa foram usadas menos recentemente. Se essas páginas tiverem sido modificadas por um processo, o Linux as considera "sujas" e tem de paginá-las para o disco antes de a memória poder ser reutilizada. As páginas que foram lavadas dessa maneira, ou que jamais ficaram sujas antes de qualquer coisa, estão "limpas" e podem ser recicladas pelo Linux para emprego em algum outro lugar.

2. Atualizar a saída de **top** muito rapidamente pode por si só ser um voraz consumidor de recursos de CPU; portanto, seja criterioso no emprego de **top**.

www.snow666.blogspot.com
Snow666

Quando uma página na lista inativa for referida por um processo, o kernel retorna seu mapeamento de memória para a tabela de páginas, reinicializa a idade da página e a transfere da lista inativa para a lista ativa. Páginas que foram gravadas em disco têm de ser trazidas para a memória antes de poderem ser reativadas. Uma "falha leve" ocorre quando um processo se refere a uma página inativa em memória, e uma "falha grave" resulta de uma referência a uma página não residente. Em outras palavras, uma falha grave requer que uma página seja lida do disco e uma falha leve, não.

A demanda por memória varia, para que o kernel possa executar **kswapd** em diferentes velocidades. Quando a demanda por memória for extrema, **kswapd** é executado com bastante freqüência, e as páginas têm de ser referidas com maior freqüência para evitar serem paginadas, pois suas idades decrescem a zero.

O sistema VM (Virtual Memory, memória virtual) depende do atraso entre o momento em que uma página é colocada numa lista inativa e o momento em que ela é efetivamente paginada para ordenar páginas ativas com base em páginas inativas. Portanto, o sistema VM tem de prever atividade de paginação futura para decidir com que freqüência deve executar **kswapd**. Se **kswapd** não for executado com freqüência suficiente, poderá ser que não haja páginas inativas limpas suficientes para satisfazer a demanda. Se **kswapd** for executado com muita freqüência, o kernel levará muito tempo processando falhas de páginas leves.

Já que o algoritmo de paginação é previsível, não há necessariamente uma correspondência um-para-um entre eventos de paginação e alocações de páginas por parte de processos em execução. O objetivo do sistema é manter memória livre suficiente para que processos não tenham efetivamente de aguardar por uma página cada vez que eles fizerem uma nova alocação. Se a paginação aumentar dramaticamente quando o sistema estiver muito ocupado, provavelmente será interessante ter mais RAM.

Se o kernel preenche tanto RAM quanto swap, toda a VM foi exaurida. O Linux usa um "matador de memória insuficiente" para tratar dessa condição. Essa função seleciona e mata um processo para liberar memória. Embora o kernel tente matar o processo menos importante em seu sistema, exaurir a capacidade de memória sempre é algo a ser evitado. Nessa situação, é provável que uma porção substancial dos recursos de sistema estejam sendo dedicados à limpeza de memória em vez de trabalho útil

Até mesmo processos sendo executados com prioridade de CPU baixa podem ser surrupiadores de páginas. Suponha, por exemplo, que você esteja rodando um cliente SETI[3] numa prioridade muito baixa (alto valor de **nice**) em sua estação de trabalho e ao mesmo tempo lendo e-mails numa janela de terminal. Ao fazer uma pausa para ler uma mensagem, o emprego de CPU cai para zero e a simulação é permitida. Ele traz todas as suas páginas, forçando o shell, o servidor de janelas, o programa para leitura de e-mails e emulador de terminal serem retirados da memória. Ao avançar para a próxima mensagem, há um atraso, já que um grande pedaço da memória do sistema é revolvido. Na vida real, um alto valor de **nice** não é garantia de que um processo não provocará problemas de desempenho.

Analisando o emprego de memória

Basicamente, três números quantificam a atividade de memória: a quantidade total de memória virtual ativa e as taxas de swapping e de paginação. O primeiro número lhe informa a demanda total por memória e os dois seguintes sugerem a proporção dessa memória que está sendo usada ativamente. O objetivo é reduzir a atividade ou aumentar a memória até que a paginação permaneça num nível aceitável. Paginação ocasional é inevitável, de modo que você não deve se preocupar sobre tentar eliminá-la completamente.

Você pode usar o comando **free** para determinar a quantidade de memória e de espaço de swap que estão efetivamente em uso. Use-o com o flag **-t** para calcular automaticamente a quantidade total de memória virtual:

```
% free -t
             total       used       free   shared  buffers  cached
Mem:        127884      96888      30996    46840    57860   10352
-/+ buffers/cache:      28676      99208
Swap:       265032       3576     261456
Total:      392916     100464     292452
```

A coluna `free` indica o número de kilobytes na lista livre do sistema; valores menores do que 3% da memória total do sistema geralmente são indicativos de problemas. Você pode usar o comando **swapon** para determinar exatamente que arquivos e partições estão sendo utilizados para espaço de swap:

```
% swapon -s
Filename         Type       Size    Used  Priority
/dev/sdb7        partition  265032  3576  -1
```

3. SETI (Search for ExtraTerrestrial Intelligence); vá ao site http://www.seti.org/science/setiathome.html.

www.snow666.blogspot.com
Snow666

Esse sistema usa uma partição de disco, sdb7, para swap. Já que a atividade de paginação pode ser um vazamento significativo em termos de largura de banda de disco, sempre é melhor, se possível, distribuir o espaço de swap por vários discos físicos. O emprego de vários discos de swap diminui o tempo necessário que um disco deve gastar lidando com dados swap e aumenta a largura de banda de armazenamento de swap.

Consulte a seção 12.4 para obter maiores informações sobre o sistema de arquivos /proc.

Em sistemas UNIX tradicionais, o comando **vmstat** fornece informações sobre paginação e swapping. Entretanto, o comando **procinfo** que vem com a maioria das distribuições Linux é uma fonte de informações melhor. Embora **procinfo** venha com todos os sistemas usados como exemplo em nosso livro, ele não é necessariamente incluído na instalação-padrão, de modo que talvez você precise adicionar o pacote **procinfo** manualmente.

procinfo não possui uma maneira especial de obter informações sobre seu sistema; ele simplesmente formata os dados dos arquivos em seu sistema de arquivos **/proc**. Sem ele, interpretar os arquivos **/proc** pode se tornar um desafio considerável. Para atualizações contínuas a cada cinco segundos, você pode executar **procinfo -n5**.

```
% procinfo
Linux 2.2.16-22 (root@porky) (gcc egcs-2.91.66) #1 Tue Aug 22 16:49:06 EDT
      2000 1CPU [redhat]

Memory:     Total   Used   Free   Shared  Buffers  Cached
Mem:        37960  30320   7640    12136     5312   14536
Swap:       81964   5236  76728

Bootup: Sat Jul 28 16:28:13 2001   Load average: 0.10 0.14 0.09 1/50 15406

user :  3d   6:56:45.34   0.0% page in  : 1408236 disk 1: 255812r 2126666w
nice :       0:00:00.00   0.0% page out : 5880596 disk 2: 345755r 1276809w
system: 1d   11:55:07.27  4.0% swap in  :   34087
idle :  32d  4:58:45.62  87.1% swap out :   17825
uptime: 36d  23:50:38.22       context  : 54303486

irq 0:   241327308 timer       irq 10: 3032801 eth0
irq 1:         538 keyboard    irq 12:    2100 PS/2 Mouse
irq 2:           0 cascade [4] irq 13:       1 fpu
irq 6:           3             irq 14: 1905415 ide0
irq 8:           2             irq 15:       5 ide1
```

Algumas das informações na saída de **procinfo** sobrepõem aquelas de **free**, **uptime** e **vmstat**. Além disso, **procinfo** fornece informações sobre sua distribuição de kernel, paginação de memória, acesso a disco e atribuições IRQ. Você pode usar **procinfo -a** para ver até mais informações de seu sistema de arquivos **/proc**, inclusive opções de boot de kernel, módulos carregáveis de kernel, dispositivos de caracteres e sistemas de arquivos.

Quaisquer inconsistências aparentes entre as colunas relacionadas com memória são em grande parte ilusórias. Algumas colunas contam páginas e outras contam kilobytes. Todos os valores são médias arredondadas. Além disso, algumas são médias de quantidades escalares e outras são deltas (variações) médios. Não poderíamos, por exemplo, calcular o próximo valor de `Free` por meio de `Free` e informações de paginação atuais pois os eventos de paginação que determinam o próximo valor médio de `Free` ainda não ocorreram.

Use os campos `page in/out` e `swap in/out` para avaliar a paginação do sistema e o comportamento de swapping. Um page-in não representa necessariamente uma página que está sendo recuperada da área de swap. Ele poderia ser um código executável sendo paginado de um sistema de arquivos ou uma página cópia-na-gravação sendo duplicada, ambos os quais são ocorrências normais que não indicam necessariamente uma falta de memória. Por outro lado, page-outs sempre representam dados sendo forçosamente ejetados pelo kernel.

Se o seu sistema possui um fluxo constante de page-outs, você precisará de mais memória. Mas se a paginação acontece somente ocasionalmente e não produz soluços irritantes ou reclamações por parte dos usuários, você poderá ignorá-la. Se seu sistema se encaixa num meio-termo, análise adicional dependerá se você está tentando otimizar para desempenho interativo (por exemplo, uma estação de trabalho) ou tentando configurar uma máquina com muitos usuários simultâneos (por exemplo, um servidor).

Se metade das operações forem page-outs, você poderá calcular que cada 50 page-outs provoca cerca de um segundo de latência. Se 75 page-outs devem ocorrer para lhe permitir rolar uma janela, você aguardará cerca de 1,5 segundos. Uma regra prática usada por desenvolvedores de interfaces é que um usuário mediano percebe que o sistema está "lento" quando os tempos de resposta forem maiores do que sete décimos de segundo.

www.snow666.blogspot.com
Snow666

Também vale a pena ressaltar que **procinfo** fornece algumas informações de CPU que não são visíveis com **vmstat** ou **uptime**. Além de relatar a média de carga em períodos de 5, 10 e 15 minutos, **procinfo** lista o número instantâneo de processos em execução, o número total de processos e o ID de processo do último processo executado. Eis, por exemplo, um trecho da saída gerada por **procinfo** num servidor atarefado:

```
Load average: 2.37 0.71 0.29 3/67 26941
```

Esse servidor possui 67 processos no total, dos quais um era executável. O último processo a ser executado tinha PID 26941 (neste caso, era o processo de shell por meio do qual **procinfo** era executado).

Analisando o I/O de disco

Podemos monitorar o desempenho do disco através do comando **iostat**. Assim como **vmstat**, ele aceita argumentos opcionais para especificar um intervalo em segundos e uma contagem de repetição, e sua primeira linha de saída é um sumário desde o momento da inicialização do sistema. Como **vmstat**, ele também o informa como o tempo de CPU está sendo gasto. Eis um rápido exemplo que omite algumas informações que não são específicas de discos:

```
% iostat
...
Disks:    tps   Blk_read/s   Blk_wrtn/s  Blk_read  Blk_wrtn
hdisk0   0.54         0.59         2.39    304483   1228123
hdisk1   0.34         0.27         0.42    140912    216218
hdisk2   0.01         0.02         0.05      5794     15320
hdisk3   0.00         0.00         0.00         0         0
```

iostat coleta informações do sistema de arquivos **/proc** para produzir uma linha de saída para cada disco físico em seu sistema. Infelizmente, o Linux mantém somente informações estatísticas mínimas sobre o disco e até mesmo as informações que são preservadas são de uso limitado. Cada disco rígido possui as colunas `tps`, `Blk_read/s`, `Blk_wrtn/s`, `Blk_read` e `Blk_wrtn`, indicando transferências de I/O por segundo, blocos lidos por segundo, blocos gravados por segundo, total de blocos lidos e total de blocos gravados. Linhas de zeros serão relatadas caso você tenha menos do que quatro discos.

Os blocos de disco têm, tipicamente, um tamanho igual a 1K, de modo que você pode rapidamente determinar o desempenho real do disco em kilobytes. Por outro lado, as transferências são definidas de maneira muito mais nebulosa. Uma solicitação de transferência pode incluir várias solicitações de I/O lógicas em relação a vários setores, de forma que estes dados também são mais úteis para identificar tendências ou comportamento irregular.

O custo de busca é o fator mais importante afetando o desempenho da unidade de disco. Para uma primeira aproximação, a velocidade rotacional do disco e a velocidade do barramento com o qual ele está conectado têm relativamente pouco impacto. Discos modernos podem transferir dezenas de megabytes de dados por segundo caso eles sejam lidos de setores contíguos, mas eles podem realizar somente 100 a 300 buscas por segundo. Se você transferir um setor por busca, poderá facilmente constatar menos do que 5% do desempenho de pico da unidade de disco.

Buscas são mais custosas quando elas fazem com que as cabeças do disco percorram uma longa distância. Se você tem um disco com várias partições de disco e os arquivos são lidos de cada partição numa ordem aleatória, as cabeças terão de ficar indo e voltando por um longo percurso para alternar entre partições. Por outro lado, arquivos dentro de uma partição serão relativamente locais entre si. Ao particionar um disco novo, talvez seja interessante considerar as implicações de desempenho e colocar arquivos que são acessados juntos no mesmo sistema de arquivos.

Para efetivamente atingir o máximo de desempenho em disco, você deve colocar os sistemas de arquivos que são usados juntos em discos diferentes. Embora dependa da arquitetura do barramento e dos drivers de dispositivos, a maioria dos computadores é capaz de gerenciar vários discos independentemente, aumentando dramaticamente o desempenho. Por exemplo, muitas vezes vale a pena distribuir dados de servidor Web freqüentemente acessados e entrar no sistema escolhendo um de vários discos.

Em sistemas que possuem mais do que alguns discos, considere utilizar discos SCSI com um controlador especializado. Essa técnica será escalável somente para uma capacidade limitada e dependerá basicamente da arquitetura de hardware utilizada. Leia a documentação sobre seus equipamentos ou consulte seu fornecedor.

É especialmente importante distribuir, se possível, a área de swap entre vários discos, já que a paginação tende a reduzir o desempenho de todo o sistema. Essa configuração é suportada através do emprego do comando **swapon**. O kernel do Linux pode usar tanto partições de swap dedicadas quanto arquivos de swap num sistema de arquivos formatado. Partições dedicadas são mais eficientes; não use arquivos de swap caso tenha uma opção.

Sistemas Linux modernos permitem que você configure vários "sistemas de arquivos baseados em memória" que são, essencialmente, a mesma coisa que discos RAM de PCs. Um driver especial atua como um disco, mas na

www.snow666.blogspot.com
Snow666

verdade armazena dados numa memória de alta velocidade. Muitas instalações usam um disco RAM para seus sistemas de arquivos **/tmp** ou para outros arquivos de alta atividade como logs de servidor Web ou spools de e-mails. Usar um disco RAM poderá reduzir a memória disponível para uso geral, porém torna a leitura e a gravação de arquivos temporários incrivelmente mais rápidas. Geralmente é um bom negócio.

A maioria dos kernels Linux foi compilada com suporte a disco RAM. Em sistemas que suportam discos RAM, o diretório **/dev** contém vários arquivos de dispositivos de disco RAM como **/dev/ram0** e **/dev/ram1**. O número de dispositivos pode variar, porém normalmente são pelo menos cinco.

Para usar um disco RAM, primeiramente formate o sistema de arquivos num dispositivo RAM não utilizado, a seguir monte-o num diretório existente:

```
# mke2fs /dev/ram12
# mount /dev/ram12 /tmp/fastdisk
```

O tamanho de RAM padrão é somente de 4MB, o que realmente não é grande o suficiente para **/tmp**. Infelizmente, o procedimento para alterar o tamanho é um tanto árduo, pois ele é configurado na forma de uma variável de kernel. Você poderá adicionar uma linha como `ramdisk_size=100000` à sua configuração de carregador de inicialização para passar o novo valor no momento do boot (o tamanho é dado em blocos de 1K) ou configurar o driver de disco RAM como um módulo que pode ser carregado dinamicamente. No último caso, você adicionaria o argumento necessário ao arquivo **/etc/modules.conf** ou o passaria na forma de um argumento para **insmod**.

Alguns pacotes degradam o desempenho do sistema retardando operações básicas. Dois exemplos são cotas de disco e contabilidade de CPU. As cotas requerem que um sumário de uso de disco seja atualizado à medida que os arquivos são gravados e apagados. A contabilidade de CPU grava um registro de dados num arquivo contábil cada vez que um processo é completado. O caching em disco ajuda a minimizar o impacto desses recursos, porém eles ainda podem ter um ligeiro efeito sobre o desempenho e não devem ser habilitados a menos que você realmente os use.

26.4 Socorro! Meu Sistema Simplesmente Está uma Lesma!

Em seções anteriores falamos principalmente sobre questões relacionadas ao desempenho médio de um sistema. Soluções para esses problemas de longo prazo geralmente assumem a forma de ajustes de configuração ou de atualizações.

Entretanto, você descobrirá que mesmo sistemas configurados apropriadamente são, algumas vezes, mais lentos do que o usual. Felizmente, problemas transientes normalmente são fáceis de serem diagnosticados. Noventa por cento das vezes, eles são provocados por um processo voraz que simplesmente está consumindo tanto poder de processamento, de disco ou largura de banda da rede que os demais processos ficam emperrados. Ocasionalmente, processos mal-intencionados consomem vorazmente recursos disponíveis para intencionalmente deixar um sistema ou rede lentos, um esquema conhecido como ataque DOS (Denial Of Service, recusa de serviço).

Normalmente você pode informar qual recurso está sendo consumido vorazmente mesmo sem executar um comando de diagnóstico. Se o sistema parecer "maluco" ou você ouvir o disco sendo acessado como louco, o problema mais provável é uma falta de memória ou de largura de banda de disco.[4] Se você estiver sentindo que o sistema está "lento" (tudo leva muito tempo e os aplicativos não podem ser "ligados"), o problema pode ser a CPU.

A primeira etapa no diagnóstico é executar **ps auxww** ou **top** para procurar processos descontrolados óbvios. Qualquer processo que estiver usando mais do que 50% de CPU provavelmente deve estar com problemas. Se nenhum processo estiver consumindo individualmente uma parcela excessiva de CPU, verifique quantos processos estão usando pelo menos 10% dos recursos. Se houver mais do que dois ou três (não conte o próprio **ps**), a carga média provavelmente deve ser bem alta. Isso é, por si só, uma causa de desempenho sofrível. Verifique a carga média com **uptime** e use **vmstat** ou **top** para ver se a CPU está alguma vez ociosa.

Se não for evidente nenhuma disputa de CPU, verifique o nível de paginação que está ocorrendo via **vmstat** ou **procinfo**. Toda a atividade em disco é interessante: um número grande de paginação para o disco pode indicar disputa de memória, e tráfego de disco na ausência de paginação pode significar que um processo está monopolizando o disco fazendo constantemente leitura e gravação de arquivos.

Não há nenhuma maneira direta de vincular operações de discos a processos, porém **ps** pode reduzir o número de possíveis suspeitos para você. Qualquer processo que estiver gerando tráfego em disco tem de estar usando algu-

4. Isto é, leva muito tempo para alternar entre aplicativos, mas o desempenho é aceitável quando um aplicativo está repetindo uma tarefa simples.

Snow666

ma quantidade de tempo de CPU. Normalmente é possível se fazer uma estimativa fundamentada sobre qual dos processos ativos é o verdadeiro culpado.[5] Use **kill -STOP** para suspender o processo e testar sua teoria.

Suponha que você ache que um determinado processo esteja com problemas — o que fazer? Usualmente, nada. Algumas operações simplesmente requerem um bocado de recursos e estão propensas a tornar o sistema lento. Isso não significa necessariamente que elas são ilegais. Normalmente é aceitável aplicar **renice** a um processo inoportuno que está vinculado à CPU. Mas certifique-se de pedir ao proprietário para usar o comando **nice** no futuro. Algumas vezes, o ajuste fino de aplicativos pode reduzir dramaticamente a demanda de um programa por recursos de CPU; este efeito é especialmente visível com software de servidor de rede personalizado como os aplicativos para Web.

Processos que são consumidores vorazes de disco ou de memória não são tão fáceis de serem tratados. **renice** geralmente não ajudará. Você tem a opção de matar ou interromper um processo, porém não recomendamos este tipo de procedimento caso a situação não seja uma emergência. Como acontece com os glutões de recursos de CPU, você pode usar a solução rudimentar de solicitar ao dono do processo para executá-lo mais tarde.

O kernel Linux permite que um processo restrinja seu próprio uso de memória física com a chamada de sistema **setrlimit**. Essa facilidade também se encontra disponível no shell C por meio do comando embutido **limit**. Por exemplo, o comando

```
% limit memoryuse 32m
```

faz com que todos os comandos subseqüentes que o usuário executar tenham sua própria memória física limitada a 32 megabytes. Esse recurso é grosseiramente equivalente ao comando **renice** para processos vinculados à memória. Talvez você possa sugerir diplomaticamente que transgressores coloquem uma linha como essa em seus arquivos **.cshrc**.

Se um processo descontrolado parece não ser a fonte de desempenho sofrível, há duas outras possíveis causas a serem investigadas. A primeira delas é uma rede sobrecarregada. Muitos programas estão tão intimamente vinculados à rede que é difícil dizer onde termina o desempenho do sistema e onde começa o desempenho da rede. Veja o Capítulo 20 para obter maiores informações sobre as ferramentas usadas para monitorar redes.

Alguns problemas de rede são difíceis de serem diagnosticados, pois eles vêm e vão rapidamente. Por exemplo, se cada máquina da rede executar um programa relacionado com rede por meio de **cron** num determinado momento do dia, normalmente haverá uma breve, porém dramática, disfunção. Toda máquina da rede ficará paralisada por cinco segundos e depois o problema desaparecerá tão rapidamente quanto surgiu.

Atrasos relacionados com servidores é uma outra possível causa de crises de desempenho. Sistemas UNIX estão constantemente consultando servidores remotos em busca de NFS, NIS, DNS e uma série de outros recursos. Se um servidor estiver morto ou algum outro problema tornar difícil a comunicação com ele, os efeitos podem se propagar através de sistemas-cliente.

Por exemplo, num sistema atarefado, algum processo poderá vir a usar a rotina de biblioteca **gethostent** a cada poucos segundos. Se uma pequena disfunção DNS fizer com que essa rotina leve dois segundos para ser completada, provavelmente você perceberá a diferença no desempenho global. Problemas de encaminhamento DNS e de configuração de pesquisa reversa são responsáveis por um surpreendente número de problemas de desempenho no servidor.

26.5 LEITURA RECOMENDADA

LOUKIDES, MIKE; GIAN-PAOLO D. MUSUMECI. *System Performance Tuning. 2. ed.*; Sebastopol: O'Reilly & Associates. 2002.

FINK, JASON R.; MATTHEW D. SHERER. *Linux Performance Tuning and Capacity Planning.* Indianapolis, IN: SAMS. 2001.

KILLELEA, PATRICK; LINDA MUI. *Web Performance Tuning: Speeding Up the Web.* Sebastopol: O'Reilly & Associates. 1998.

26.6 EXERCÍCIOS

E26.1 Faça uma estimativa fundamentada no que se refere ao problema em cada um dos seguintes casos:

a) Ao alternar entre aplicativos, o disco é utilizado excessivamente e há um retardo perceptível.

---

5. Um espaço de endereços virtual grande ou conjunto residente costumava ser um sinal de suspeita, porém bibliotecas compartilhadas tornaram estes números menos úteis. **ps** não é muito inteligente em relação à separação dos tamanhos das bibliotecas de sistema compartilhadas do espaçao de endereços de cada processo individual. Muitos processos parecem erroneamente ter megabytes de memória ativa.

Snow666

b) Um programa de simulação numérica leva mais tempo do que o normal, porém a memória do sistema está em grande parte livre.
c) Usuários numa rede local muito ativa reclamam de acesso NFS lento, porém a média de carga no servidor está muito baixa.
d) A execução de um comando (qualquer comando) normalmente indica "memória insuficiente".

E26.2 Execute **procinfo** numa máquina Linux disponível e discuta os resultados obtidos. Quais, se algum, recursos parecem estar sendo usados pesadamente? Que recursos parecem não estar sendo utilizados? Inclua a listagem IRQ em sua análise.

★ **E26.3** O balanceamento de carga pode ter um impacto dramático no desempenho do servidor conforme visto do mundo externo. Sem mencionar detalhes, discuta vários mecanismos que poderiam ser usados para realizar o balanceamento de carga.

★ **E26.4** Liste os quatro recursos principais que afetam o desempenho. Para cada recurso, dê um exemplo de um aplicativo que poderia facilmente levar à exaustão deste recurso. Discuta maneiras de aliviar parte do esforço associado a cada situação.

★ **E26.5** Usando a Web e/ou as páginas de manual, procure o comando **hdparm**. Que opções ele oferece para testar velocidades de acesso a disco? Como **hdparm** pode aumentar as velocidades de acesso a disco em alguns casos?

★★ **E26.6** Selecione dois programas que usem uma quantidade de recursos de sistema visível. Use **vmstat** e as demais ferramentas mencionadas no capítulo para dar um perfil de ambos os aplicativos. Faça uma argumentação sobre o que cada programa faz que o torna um voraz consumidor de recursos. Baseie suas afirmações em dados.

Snow666

27 Cooperando com o Windows

É fato concreto que o Windows não está saindo de cena. Não podemos ignorá-lo. Temos de coexistir pacificamente com ele. Uma maneira de encarar a situação é que o UNIX e o Linux trouxeram o TCP/IP e a Internet para a mesa, enquanto o Windows trouxe milhões de usuários. O desafio para os administradores de sistema é ser o anfitrião deste jantar nefasto.[1]

Felizmente, a medicina moderna (ferramentas administrativas) pode reduzir significativamente a probabilidade de uma indigestão causada pelo Windows. É verdade que ambas as plataformas têm seus pontos fortes e que elas podem ser usadas para que cooperem entre si. O Windows é uma plataforma desktop popular e cheia de recursos, capaz de cobrir a lacuna deixada entre o usuário e o cabo de rede que vem da parede. Por outro lado, o Linux é uma plataforma de infra-estrutura confiável e expansível.

Este capítulo cobre uma série de tópicos com os quais os administradores se deparam nesta era pós-moderna em que PCs e UNIX convivem.

27.1 Compartilhamento de Arquivos e de Impressoras

Talvez o nível mais poderoso da integração Windows/Linux seja atingido pelo compartilhamento de diretórios que residem num host Linux (ou um servidor de arquivos dedicado tipo Linux) com PCs desktop que rodam Windows. Os diretórios compartilhados podem ser dispostos de tal forma que apareçam transparentemente sob o Windows, seja como drives ou como extensões da árvore de arquivos de rede do Windows. Você também pode montar sistemas de arquivos residentes no Windows sob o Linux, mas este arranjo é menos comum. O NFS ou CIFS podem ser usados para implementar qualquer um desses sistemas de compartilhamento de arquivos.

Veja o Capítulo 17 para obter maiores informações sobre o NFS.

NFS: Network File System

O NFS foi desenvolvido para compartilhar arquivos entre hosts UNIX, em que os paradigmas segurança e bloqueio de arquivos são significativamente diferentes daqueles do Windows. Embora uma grande variedade de produtos que montam diretórios compartilhados NFS em clientes Windows esteja disponível, seu emprego deve ser evitado a qualquer custo tanto devido à discordância de paradigmas quanto pelo fato de o CIFS simplesmente funcionar melhor. Se estiver amarrado a servidores de arquivos Windows, é razoável usar software de servidor Windows NFS para compartilhar sistemas de arquivos Windows com clientes Linux. Entretanto, o sistema de arquivos Linux SMBFS descrito a seguir normalmente é uma maneira mais fácil de atingir o mesmo objetivo.

1. Como diz o ditado: "Quem se acomuna com o inimigo tem de se precaver".

www.snow666.blogspot.com
Snow666

CIFS

O CIFS (Common Internet File System, sistema de arquivos comuns Internet) se baseia em protocolos outrora conhecidos como SMB (Server Message Block, bloco de mensagens de servidor). O SMB foi uma extensão que a Microsoft acrescentou ao DOS em seus primórdios para permitir que I/O de disco pudesse ser redirecionado a um sistema conhecido como NetBIOS (Network Basic Input/Output System, sistema básico de entrada/saída para redes). Desenvolvido pela IBM e Sytec, o NetBIOS era uma interface rudimentar entre a rede e um aplicativo.

No mundo moderno, os pacotes SMB são transportados numa extensão do NetBIOS conhecida como NBT, NetBIOS sobre TCP. Embora isso pareça muito enrolado, o resultado é que esses protocolos se tornaram amplamente usados e se encontram disponíveis em plataformas que vão desde MVS e VMS para nossos caros amigos Linux e Windows. Todo o mundo agora dança.

SMBFS: monte compartilhamentos CIFS remotos

O sistema de arquivos SMBFS permite que o Linux trate servidores de arquivos CIFS de rede da mesma maneira que um outro tipo de sistema de arquivos externo, de modo muito parecida como ele trata servidores NFS. O SMBFS vem embutido no kernel Linux há vários anos. Originalmente escrito por Pal-Kristian Engstad, o código agora é mantido pelo autor do Samba, Andrew Tridgell. Pelo fato do sistema de arquivos ser embutido no kernel, você pode usar o bom e velho comando **mount** para conectar um compartilhamento CIFS remoto a um ponto de montagem local. Eis um exemplo simples:

```
# mount -t smbfs -o username=ned //engserver/admin /home/ned/engadmin
```

O parâmetro **-t smbfs** informa **mount** que ele deve usar o código de sistema de arquivos SMBFS para montar o compartilhamento CIFS. Especificamos um nome de usuário após a opção **-o**. Finalmente, informamos **mount** para procurar um serviço "admin" no servidor "engserver" e conectá-lo ao diretório local existente **~ned/engadmin**.

Você pode especificar muitas outras opções de montagem específicas ao CIFS após o flag **-o** para montagens SMBFS, inclusive opções de depuração, de grupos de trabalho e senhas de login. A página de manual para **smbmount** lista todas as opções. Entretanto, recomendamos que você não use diretamente o comando **smbmount**, já que **mount** fornece uma interface consistente e mais fácil de se lembrar (**mount** chama **smbmount** para você).

Samba: servidor CIFS para Linux

O Samba é um pacote de software tremendamente popular, disponível sob a licença pública GNU, que implementa o lado servidor do CIFS em hosts Linux. Ele foi criado originalmente por Andrew Tridgell, um australiano, que aplicou engenharia reversa sobre o protocolo SMB e publicou o código resultante em 1992.

Hoje o Samba tem bom suporte e está sendo atualizado ativamente para expandir sua funcionalidade. Ele fornece um mecanismo estável de poder industrial para integrar máquinas Windows numa rede Linux. A real beleza do Samba é que você precisa somente instalar um pacote na máquina Linux; não é necessário nenhum software especial no lado Windows.[2]

O CIFS provê cinco serviços básicos:

- Compartilhamento de arquivos
- Impressão em rede
- Autenticação e autorização
- Determinação de nomes
- Anúncio de serviços ("navegação" de impressoras e de servidor de arquivos)

O Samba não somente serve arquivos Linux via CIFS, como também pode realizar todas as funções básicas de um controlador de domínio primário Windows NT 4.0. O Samba suporta alguns recursos avançados, inclusive logins de domínios Windows NT, perfis de usuários móveis ("roaming") Windows e spooling de impressão CIFS. O Samba facilita o compartilhamento de arquivos entre o Linux e toda versão do Windows, do Windows 95 em diante.

A Microsoft não compartilha informações sobre como ele implementa tecnologias como o CIFS, domínios NT e Active Directory, de modo que o pessoal engenhoso do projeto Samba tem realmente de aplicar engenharia reversa em servidores rodando Microsoft para descobrir como fazer com que o Samba funcione. Infelizmente, essa restrição introduz uma defasagem antes de novos recursos da Microsoft poderem ser implementados no Samba. Na época em que este livro era escrito, faltava no Samba a maioria dos recursos de controlador de domínio Windows

2. Desde que a máquina Windows 95 já tenha sido configurada para "rede Microsoft".

Snow666

2000, como Kerberos, Active Directory, confiabilidade de domínios e replicação com controladores de domínio NT. O Samba é uma solução de primeira categoria para um grande número de organizações; entretanto, os recursos avançados talvez não funcionem tão bem quanto em redes Windows 2000 ou Active Directory complexas.

A maior parte da funcionalidade do Samba é implementada por dois daemons: **smbd** e **nmbd**. **smbd** implementa serviços de arquivos e de impressão, bem como autenticação e autorização. **nmbd** fornece os demais componentes CIFS principais, determinação de nomes e anúncio de serviços.

Diferentemente do NFS, que é profundamente entremeado com o kernel, o Samba não requer nenhuma modificação no kernel e é executado totalmente como um processo de usuário. Ele se vincula aos soquetes usados para solicitações NBT e aguarda que um cliente solicite acesso a um recurso. Assim que a solicitação tiver sido feita e autenticada, o **smbd** cria uma instância de si mesmo que é executada como o usuário que está fazendo as solicitações. Conseqüentemente, todas as permissões de acesso a arquivos Linux (inclusive permissões de grupos) são obedecidas. A única funcionalidade especial que o **smbd** acrescenta a isso é um serviço de bloqueio a arquivos que dá aos PCs clientes a semântica de bloqueio com os quais eles estão acostumados.

Instalando e configurando o Samba

O Samba vem com todas as distribuições do Linux vistas neste livro. Correções, documentação e outros adendos se encontram disponíveis em www.samba.org. Versões mais antigas do que 2.2.0a têm brechas de segurança conhecidas e devem ser atualizadas imediatamente.

Em todos os sistemas, você precisará editar o arquivo **smb.conf** (**/etc/samba/smb.conf** no Red Hat e Debian, **/etc/smb.conf** no SuSE) para informar ao Samba como ele deve se comportar. Nesse arquivo, você especifica os diretórios e impressoras que devem ser compartilhados, seus direitos de acesso e parâmetros operacionais genéricos do Samba. Todas as opções são documentadas na página de manual **smb.conf**, que você realmente precisa consultar ao integrar o Samba numa rede na qual o compartilhamento de arquivos Microsoft já está em uso. O pacote Samba vem com um modelo de arquivo **smb.conf** bem comentado que geralmente é um excelente ponto de partida para configurações novas.

É importante atentar às implicações de segurança no compartilhamento de arquivos ou recursos através de uma rede. O Samba possibilita um controle refinado sobre a segurança, mas ele somente funciona se você o utilizar efetivamente. Para uma instalação típica, você precisa de duas coisas para garantir um nível de segurança básico:

- No arquivo **smb.conf**, a estrutura `hosts allow` controla quais clientes podem acessar os recursos compartilhados pelo Samba. Certifique-se de que ele contenha somente os endereços IP (ou intervalos de endereços) que ele deve ter.
- Você deve *obrigatoriamente* bloquear o acesso pela Internet a portas TCP CIFS com um firewall para filtragem de pacotes. Estas são as portas TCP 137 a 139. Maiores informações sobre como fazer isso são dadas na seção 21.10.

Eis um exemplo de um arquivo **smb.conf** completo para uma rede simples:

```
[global]

# grupo de trabalho = Nome-de-Domínio-NT ou Nome-de-Grupo-de-Trabalho

  workgroup = MYGROUP

# Lista os hosts que podem acessar objetos compartilhados via Samba.
# Aqui, somente hosts em duas redes classe C são permitidos.

  hosts allow = 192.168.1. 192.168.2.

# Carrega automaticamente sua lista de impressoras de um arquivo.[3]

  printcap name = /etc/printcap
  load printers = yes

# Usa um arquivo de log distinto para cada máquina e limita o tamanho a 500K cada.

  log file = /var/log/samba/log.%m
```

3. As impressoras já têm de estar configuradas no host Linux. Veja o Capítulo 24 para obter maiores detalhes.

www.snow666.blogspot.com
Snow666


```
  max log size = 500

# Configura o modo de segurança. A maioria das pessoas irá querer nível de segurança usuário. Veja
# o arquivo security_level.txt na documentação do Samba para maiores detalhes.

  security = user

# Recomendamos que você use criptografia de senhas. Favor ler os arquivos
# ENCRYPTION.txt, Win95.txt e WinNT.txt na documentação do Samba.
# Não habilite esta opção a menos que tenha lido estes documentos.

; encrypt passwords = yes
; smb passwd file = /etc/smbpasswd

# A maioria das pessoas achará que esta opção resulta num melhor desempenho. Leia
# speed.txt e as páginas de manual para detalhes.

  socket options = TCP_NODELAY

# Compartilha diretórios domésticos. Por exemplo, ~trent sob o Linux aparecerá
# como "trent" compartilhado no navegador PC.

[homes]
   comment = Home Directories
   browseable = no
   writable = yes

# Compartilha todas as impressoras.

[printers]
   comment = All Printers
   path = /var/spool/samba
   browseable = no
   guest ok = no
   writable = no
   printable = yes

# Compartilha um diretório específico.

[devel]
   comment = Staff Development Shared Directory
   path = /devel/shared
   public = no
   writable = yes
   printable = no
   create mask = 0775
```

Como pode ser observado pelos comentários, o arquivo **smb.conf** é configurado de modo que, quando os usuários entrarem em seus PCs, seus diretórios domésticos e o diretório **/devel/shared** fiquem ambos disponíveis. Eles também podem imprimir para todas as impressoras que o servidor conhece.

Depurando o Samba

O Samba normalmente roda sem requerer muita atenção. Entretanto, se você tiver um problema, poderá consultar duas fontes primárias para informações de depuração: os arquivos de log por cliente e o comando **smbstatus**.

A localização de arquivos de log é especificada no arquivo **smb.conf**, e neste diretório você localizará um arquivo para cada cliente que tentou estabelecer uma conexão. **smbd** arruma esses arquivos para que eles não excedam seus tamanhos máximos especificados.

As entradas de log resumidas a seguir mostram uma série de conexões bem-sucedidas depois de uma senha digitada incorretamente:

```
[2002/01/13 11:48:24] smbd/reply.c Rejecting user 'ned': authentication failed
[2002/01/13 11:48:26] smbd/service.c klum (192.168.1.51) connect to service ned
```

Snow666


```
      as user ned (uid=500, gid=500) (pid 1275)
[2002/01/13 13:04:19] smbd/service.c klum (192.168.1.51) closed connection to
      service ned
[2002/01/13 13:04:20] smbd/service.c klum (192.168.1.51) connect to service
      admin as user ned (uid=500, gid=500) (pid 1275)
```

smbstatus permite que você examine conexões ativas e arquivos abertos no momento. Essas informações podem ser especialmente úteis quando você estiver rastreando problemas de bloqueios ("Qual usuário tem o arquivo **xyz** aberto de forma exclusiva para leitura-gravação?"). A primeira parte da saída lista os recursos com os quais um usuário se conectou. A segunda parte lista quaisquer bloqueios de arquivos ativos e a terceira parte exibe o emprego de recursos de **smbd**.[4]

```
Samba version 2.0.7
Service   uid   gid    pid  machine
----------------------------------
ned       ned   ned   1275  klum    (192.168.1.51)
goldmine  trent trent 1279  adminpc (192.168.1.50)
admin     ned   ned   1275  klum    (192.168.1.51)

Locked files:
Pid   DenyMode   R/W     Oplock Name
--------------------------------------------------------------------
1275  DENY_WRITE RDONLY  EXCLUSIVE+BATCH /home/admin/doc/ips.txt
1275  DENY_NONE  RDWR    NONE            /home/ned/install_notes.txt

Share mode memory usage (bytes):
   1048336(99%) free + 168(0%) used + 72(0%) overhead = 1048576(100%) total
```

Quando aplicativos do Windows travam, eles normalmente deixam arquivos bloqueados provocando, conseqüentemente, problemas quando o aplicativo for reiniciado. Você pode forçar que um bloqueio seja liberado matando o **smbd** "dono" do bloqueio; o PID para cada bloqueio é mostrado na saída gerada por **smbstatus**. Não diga aos seus usuários que você sabe como fazer isso e esteja atento que romper bloqueios pode induzir a arquivos corrompidos. Algumas vezes você tem de fazer o que tem de ser feito.

Consulte a seção 18.3 para obter maiores informações sobre a chave de serviço de nomes.

Autenticação unificada de logins no Windows e Linux

As versões mais recentes do Samba vêm com uma ferramenta opcional chamada Winbind. Plugando nos módulos de biblioteca de baixo nível de comutação de serviço de nomes ("name service switch"), o Winbind permite que um servidor Linux obtenha suas informações de usuário e de grupos de um controlador de domínios Windows. O Winbind também permite que usuários alterem suas senhas de domínio Windows por meio do Linux. Além disso, Winbind inclui um módulo PAM (veja a seção 21.3 para maiores informações sobre o PAM) que usa o domínio Windows para autenticar logins Linux locais. Embora Winbind seja um componente relativamente novo do conjunto de ferramentas do Samba, ele poderia ser uma solução ideal para acoplar uns poucos computadores Linux numa rede predominantemente Windows.

27.2 Emulação de Terminais Segura com o SSH

Alguns usuários podem estar querendo deixar o Windows para trás e se dirigir para as encostas cobertas de neve de uma boa sessão **bash** ou shell C. Obviamente, a maneira mais fácil de conseguir isso é usar o programa **telnet** que a Microsoft comercializa junto com o Windows. Infelizmente, falta a ele algumas coisas fundamentais como recortar e colar. Da mesma forma, como acontece com qualquer implementação do protocolo TELNET, ele não tem nenhum conceito de segurança. (Mas afinal de contas, quem vai precisar de uma linha de comandos?) Felizmente, uma série de emuladores de terminais se encontra à disposição para o Windows que são significativamente mais confortáveis do que o **telnet** da Microsoft.

Consulte a seção 21.9 para obter maiores informações sobre o SSH.

Nosso emulador de terminais favorito é o SecureCRT da VanDyke Technologies, Inc. Esse produto comercial, mas barato, suporta SSH 1 e 2, TELNET, **rlogin** e consoles seriais. Ele provê quatro cifradores diferentes para criptografia de dados e também pode

4. A saída gerada por **smbstatus** contém algumas linhas muito longas; nós a condensamos para fins de clareza.

www.snow666.blogspot.com
Snow666

fornecer encaminhamento de portas para X11 e outros aplicativos como e-mail. Aprenda mais sobre SecureCRT em www.vandyke.com.

Um outro emulador comercial é o Windows SSH client produzido pela F-Secure Corporation. Informações sobre seus produtos estão disponíveis em www.fsecure.com.

Se estiver procurando um emulador gratuito, sugerimos utilizar o PuTTY. PuTTY é simples e eficiente e também vem com clientes Windows **scp** e **sftp**. Você poderá encontrar o PuTTY em

http://www.chiark.greenend.org.uk/~sgtatham/putty/

Uma outra alternativa é o TeraTerm, um competidor que segue de perto o PuTTY. Embora o emulador Tera-Term não venha com suporte SSH como padrão, é fácil adicionar esta funcionalidade com o plug-in TTSSH. Você poderá encontrar o TeraTerm em

http://hp.vector.co.jp/authors/VA002416/teraterm.html

O plug-in reside em

http://www.zip.com.au/~roca/ttssh.html

Juntas, essas ferramentas fornecem um ambiente bem razoável e seguro.

27.3 EMULADORES X WINDOWS

X Windows é um sistema de janelamento que de maneira nenhuma está relacionado com os sistemas operacionais Windows da Microsoft. O Sistema X Window foi desenvolvido no MIT nos anos 80 e foi adotado pelos fabricantes de estações de trabalho UNIX como padrão de ambiente com janelas (algumas vezes, com modificações substanciais). Não se preocupe, há emuladores X.

Os emuladores X Windows operam através da implementação do protocolo X11 num PC Windows. O servidor fornece um conduíte entre aplicativos-cliente (como o emulador de terminais **xterm**) e o PC Windows. Assim que o servidor X11 tiver sido iniciado no PC, clientes poderão ser executados nos ambientes UNIX ou Linux e exibidos no desktop Windows. Para aqueles que viveram o início da década de 90, isso basicamente transforma uma máquina Windows num terminal X a base de esteróides. Aplicativos X exibidos dessa maneira podem coexistir com outros aplicativos Windows no desktop. Com alguns emuladores, também é possível usar um gerenciador de janelas no ambiente Linux.

Há dezenas de emuladores X disponíveis. Os dois que impressionaram foram o X-ThinPro da Labtam (www.labtam-inc.com) e o SuperX da Frontier Technologies (www.frontiertech.com). Usamos bastante os dois sem nenhum problema. O Labtam oferece uma versão melhorada chamada X-SecurePro que agrega software de tunelamento SSH seguro. O SuperX faz um bom trabalho de garantir que o recorte e colagem entre aplicativos X e Windows funcione corretamente, e sua associação de fontes é particularmente fácil.

27.4 CLIENTES DE E-MAIL EM PCs

A primeira coisa que os usuários querem fazer em seus PCs é verificar e-mails. Fornecer um ambiente de e-mail estável é essencial para a maioria das organizações; felizmente, esta é uma área na qual os PCs Windows em desktops e os servidores centralizados Linux realmente brilham juntos.

Os clientes de e-mail Windows como o Microsoft Outlook, Netscape Messenger e o Eudora da Qualcomm são empacotados com recursos muito atraentes e de longe ofuscam os programas para e-mail UNIX do passado. Eles permitem que os usuários troquem e-mails comuns, e-mails criptografados, e-mails com anexos e e-mail com texto especialmente formatado ou até mesmo texto colorido. Essas são ferramentas essenciais do mundo da Internet — estão longe os dias do **/usr/ucb/mail** e outras ferramentas de e-mail somente para texto.

Consulte a seção 19.3 para obter maiores informações sobre IMAP e POP.

Organizações precisam estar aptas a fornecer serviços de e-mails para centenas ou milhares de usuários. É aí onde o Linux entra em ação. O Linux fornece um ambiente seguro, expansível e configurável para a recepção e transmissão de e-mails na Internet. As mensagens podem ser armazenadas num servidor Linux e acessadas por clientes de e-mail Windows via protocolos criptografados SSL como IMAPS e POPS. Isso é o melhor de todos os mundos.

Uma outra vantagem dessa abordagem, especialmente quando você está usando IMAP, é que os e-mails são armazenados no servidor. Se a máquina Windows travar, as pastas de e-mails do usuário não serão perdidas. O IMAP também permite que os usuários acessem suas mensagens de uma série de localizações, como de casa ou de um quiosque quando estiverem em viagem. Há toneladas de clientes IMAP diferentes no mercado, inclusive gateways HTTP-para-IMAP baseados na Web.

www.snow666.blogspot.com
Snow666

Se a segurança de e-mail for tão importante assim para sua organização, é crítico usar serviços de criptografia SSL como IMAPS, POPS e HTTPS para proteger senhas de usuário. Além disso, é uma boa idéia habilitar filtragens adicionais que são disponíveis para todo servidor SMTP Linux popular; elas fazem um bom trabalho na captura de vírus de e-mail do Windows.

Veja o Capítulo 10 para obter informações genéricas sobre backups.

27.5 Backups em PCs

Fazer backup dos dados em PCs desktop pode ser um problema formidável, especialmente agora que a capacidade de armazenamento típica em desktop ultrapassa 20GB. Há uma série de abordagens para este problema, inclusive ferramentas de backup de rede profissionais de fabricantes como Veritas e IBM. Obviamente, sempre há a alternativa prática de usar uma unidade de fitas local. Se você tiver dinheiro e paciência, esta é uma área onde os produtos comerciais estão realmente muito acima do resto.

Mas e o restante de nós? É possível fazer um backup do disco de uma máquina Windows (todo ou parte dele) para um servidor Linux com o utilitário **smbtar** incluso no pacote Samba. **smbtar** usa **smbclient** do Samba para se conectar a um compartilhamento CIFS remoto e colocá-lo em um arquivo **tar** local. Infelizmente, essa abordagem é muito onerosa em termos de manutenção e requer tipicamente scripting personalizado. Portanto, não a recomendamos.

A melhor solução aqui parece optar por *não* fazer o backup de Windows PCs. Instalações que não fazem backups de PCs cuidadosamente doutrinaram seus usuários e configuraram software PC de modo que todos os arquivos importantes são armazenados numa unidade de rede compartilhada. Além disso, fazer com que os usuários salvem seus arquivos numa unidade de rede é importante ter certeza de que cada aplicativo seja configurado por armazenar seus dados insubstituíveis (como arquivos de clientes de e-mail, chaves criptográficas, perfis de janelas, agendas e preferências de aplicativos) numa unidade de rede. Os PCs individuais poderão então ser configurados de maneira idêntica (em termos de aplicativos instalados, configuração de desktop etc.) em toda a empresa.[5] Se um PC falhar, um outro pode substituí-lo somente com alguns minutos de trabalho. Loucura? Pode ser. Recurso inteligente? Sim!

27.6 Inicialização Dual

Gratidão aos geeks. Já lhe ocorreu "obter o máximo dos dois mundos" juntando o poder de mais um sistema operacional em seu PC. Felizmente, é possível fazer uma "inicialização dual,"[6] significando que você pode escolher um de vários sistemas operacionais no momento do boot. Linux com Windows é uma combinação popular, especialmente entre usuários desktop que têm de ficar alternando entre ambientes. Leia tudo sobre como ajustar uma configuração de inicialização dual na página 16.

Montando sistemas de arquivos externos com o Linux

O Linux vem com suporte para uma série enorme de sistemas de arquivos, de forma que você pode montar e usar partições de discos locais que foram formatadas por outros sistemas operacionais. O comando **mount** do Linux sabe como montar sistemas de arquivos Windows como FAT, FAT32 e NTFS; tudo o que você precisa saber é qual dispositivo está associado à partição Windows que deseja montar. Partições FAT e FAT32 são especificadas com o argumento **-t vfat**, e partições NTFS usam o argumento **-t ntfs** em seu lugar. Na época em que este livro era escrito, o suporte ao NTFS no Linux era seguro apenas no modo somente leitura. O software que possibilita que o Linux grave em sistemas de arquivos NTFS ainda se encontra em versão beta.

O exemplo a seguir monta a unidade "C" FAT32 Windows localizada em **/dev/hda3** no diretório Linux **/mnt/windows**:

```
# mount -t vfat /dev/hda3 /mnt/windows
```

27.7 Executando Aplicativos Windows sob o Linux

Muitas versões do Linux podem executar aplicativos Windows ... mais ou menos. Isso é feito de diversas formas, mas todas geralmente acabam se resumindo a criar algum tipo de "ambiente de máquina virtual" que, para o aplicativo,

5. Dois produtos que podem ajudar nesse processo são o Norton Ghost da Symantec e o Drive Image Pro da PowerQuest. Um arquivo-imagem de disco padrão pode ser mantido na rede ou gravado num CD-R de emergência.
6. O termo "inicialização dual" ou "dual boot" não é estritamente correto, já que pode haver mais de dois sistemas operacionais; o termo foi mantido para maior fidelidade ao original e ao uso comum. A idéia subjacente é a possibilidade de seleção de qual sistema operacional inicializar (N.R.T.).

www.snow666.blogspot.com
Snow666

parece com o velho e bom Windows. Esses ambientes virtuais são tipicamente um tanto frágeis, mas eles tendem a funcionar relativamente bem para a maioria dos principais aplicativos.

O produto comercial VMware (www.vmware.com) transforma seu PC numa máquina virtual com a capacidade de rodar vários sistemas operacionais ao mesmo tempo. O VMware roda máquinas virtuais convidadas sobre um sistema operacional de host, que tanto pode ser o Linux como o Windows NT/2000/XP. Independentemente do sistema operacional, você pode instalar como convidados tantos sistemas operacionais compatíveis com Intel quanto queira. Já que você pode rodar várias máquinas virtuais (e, portanto, vários sistemas operacionais) ao mesmo tempo, o VMware é uma ferramenta ideal para ambientes de desenvolvimento ou de testes. Ele até vem com um recurso que conecta em rede seus sistemas operacionais para que eles possam se comunicar entre si e compartilhar sua conexão Ethernet real.

Wine (www.winehq.com) fornece uma API do Windows no ambiente Linux, permitindo que você rode aplicativos Windows sobre Linux e X. Esse software gratuito converte chamadas API nativas do Windows em seus equivalentes Linux e pode fazer isto sem usar qualquer código Microsoft. O Wine oferece suporte para redes TCP/IP, dispositivos seriais e saída de som. Infelizmente, ele suporta um número muito pequeno de drivers de hardware. O projeto Wine foi iniciado em 1993 por Bob Amstadt. Entretanto, na época em que este livro era escrito, o mantenedor atual do Wine, Alexandre Julliard, afirmava que o "Wine ainda estava em desenvolvimento e não era adequado para uso em geral".

Win4Lin é uma alternativa comercial para o Wine da NeTraverse. O Win4Lin afirma ser mais estável do que o Wine e de suportar alguns aplicativos Windows a mais. Entretanto, ele requer modificações no kernel que o Wine não precisa. O Win4Lin se encontra disponível em www.netraverse.com.

Finalmente, a Sun lançou graciosamente o StarOffice, um pacote tipo Microsoft Office para Linux e Solaris, isento de pagamento. Ele inclui ferramentas comerciais básicas como uma planilha, um processador de textos e um aplicativo para banco de dados simples. Essas ferramentas podem ler e gravar arquivos gerados no Microsoft Word, Excel e PowerPoint. Veja http://www.sun.com/products/staroffice para obter maiores informações.

Depois de o código-fonte do StarOffice ter sido liberado pela Sun, ele foi adotado pela comunidade de Software Livre mundial e batizado de OpenOffice. A versão atual do StarOffice (StarOffice 7.0) é feita por meio do código-fonte do OpenOffice e é destinada principalmente ao mercado corporativo. A versão atual do OpenOffice (OpenOffice 1.1) é de excelente qualidade, podendo substituir outros pacotes de aplicativos para escritório com qualidade igual ou superior. O OpenOffice pode ser encontrado em http://www.openoffice.org, e a versão localizada (português do Brasil) em http://www.openoffice.org.br

Linux (e UNIX) no Windows

É possível transformar uma máquina desktop Windows num ambiente Linux relativamente amigável. Conforme mencionado anteriormente, o VMware é uma excelente maneira de rodar Linux sobre uma instalação Windows existente. Em muitos casos, também é relativamente fácil portar serviços ou aplicativos Linux simples para a plataforma Windows. Uma extensa lista de software Linux e UNIX agora é executada no Windows, inclusive Apache, Perl, BIND, PHP, VIM, MySQL, Ethereal e Python.

Os usuários Windows que são saudosos de uma linha de comando UNIX devem fazer o download do ambiente Cygwin em www.cygwin.com. Esse pacote, distribuído sob a Licença Pública GNU, contém um amplo complemento de comandos UNIX comuns, bem como biblioteca para portar código que implementa APIs POSIX sob o Windows. A maneira da Cygwin reconciliar as convenções de linha de comandos e sistema de arquivos do UNIX e do Windows é bem bolada e é capaz de trazer vários itens básicos de um shell UNIX para comandos Windows nativos. Além de fazer com que os usuários Linux se sintam em casa, a Cygwin facilita a execução de software Linux e UNIX sob o Windows. Veja o site Web para detalhes.

O MKS Toolkit é uma alternativa comercial para o Cygwin. Ele inclui literalmente centenas de linguagens de scripting, servidores de rede, utilitários e shells UNIX. Veja o site Web da MKS em www.mkssoftware.com para obter maiores informações.

27.8 Dicas de Hardware

Uma das coisas realmente incríveis em relação ao hardware de um PC é que ele normalmente é barato. Infelizmente, o hardware de um PC não vem de um planeta alienígena onde o ditado é "Pague menos, leve mais!". Em vez disso, como muitas coisas aqui na Terra, você recebe pelo que pagou. Mas há algumas pegadinhas para se observar, mesmo quando você estiver pagando pelo recurso legal extra.

Snow666

Primeiramente, se estiver planejando rodar num PC um sistema operacional que não seja o Windows, certifique-se de ver quais são os dispositivos de hardware suportados. Os fabricantes de dispositivos normalmente fornecem drivers Windows para todos os seus novos e engenhosos brinquedos, mas estes não lhe serão de valia no Linux. Alguns fabricantes estão começando a "pegar a idéia" e começam a distribuir drivers Linux. Em particular, laptops totalmente compatíveis com Linux finalmente foram disponibilizados ao longo dos últimos anos.

Em segundo lugar, tenha em mente que o desempenho se baseia numa série de fatores. Nos dias de hoje, a velocidade do relógio do processador não é mais o fator limitante. Normalmente em primeiro lugar será o desempenho de I/O que fará com que seu sistema se arraste. Escolha o hardware que suporte os barramentos e taxas de transferência de maior desempenho e mais recentes. Preste atenção especial ao comprar dispositivos como controladores de disco; certifique-se de que eles foram projetados para sistemas operacionais multiusuários verdadeiros e que são capazes de processar mais do que um "comando" simultaneamente. Em vez de sair à caça da máxima capacidade em GB por real, procure discos que possuam boas garantias, altas taxas de transferência e altas velocidades de rotação.

Uma outra pegadinha são modems baratos que requerem software no computador principal para realizar algum processamento de sinais. Alguns destes WinModems são suportados pelo Linux, mas vários não são. Já que é pouco provável que os fabricantes destes WinModems liberarão informações técnicas suficientes sobre seus produtos para serem portados para Linux, é melhor optar por modems que realizem todo o seu próprio processamento. Algumas placas de rede, vídeo e de som também se enquadram nessa categoria. A melhor opção é verificar a lista de compatibilidade de hardware de sua distribuição antes de escolher qualquer hardware novo.

Por fim, esteja atento que sistemas pré-empacotados de lugares como CompUSA normalmente são projetados e otimizados para executarem o sistema operacional que é instalado de fábrica, diretamente em chips especiais contidos na placa-mãe. Essas máquinas normalmente não são as melhores candidatas para reinstalação com alguma distribuição Linux e para serem usadas para hospedar o seu site Web de comércio eletrônico. Considere a aquisição de hardware de PC de qualidade de fabricantes de ponta que vendem hardware certificado para Linux, como a IBM e a Dell. Além disso, muitos fornecedores pequenos especializados fabricam computadores Linux muito bons. O site a seguir tem uma lista útil de tais fornecedores:

http://www.linuxhardware.net/cgi-bin/viewvendor.cgi

Todos esses fabricantes oferecem hardware industrial com fontes de alimentação redundantes, ventiladores extras, drives que podem ser trocados sem ter de desligar o equipamento etc. e dão garantia de que seus sistemas funcionam com diversas distribuições Linux. Além disso, eles normalmente oferecem contratos de manutenção generosos.

27.9 Leitura recomendada

Eckstein, Robert; David Collier-Brown; Peter Kelly. *Using Samba*. Sebastopol: O'Reilly & Associates. 1999.

27.10 Exercícios

E27.1 Como o Linux ganha acesso a compartilhamentos CIFS como cliente? Como o Linux atua como um servidor de compartilhamento CIFS?

E27.2 Por que você iria querer bloquear acesso Internet às portas 137–139 num servidor Samba?

★ **E27.3** Instale um dos programas emuladores que permitem que você rode aplicativos Windows num computador Linux. Experimente um aplicativo Windows. A funcionalidade está OK? E o desempenho? Você é capaz de citar um aplicativo que se recuse a rodar? Por que ele não rodaria?

★★ **E27.4** No laboratório, compare o desempenho de um cliente que acesse arquivos através do Samba com um que acesse arquivos de um servidor CIFS nativo (isto é, uma máquina Windows). Se seus dois servidores de teste tiverem hardware diferente, imagine uma maneira de ajustar a diferença de hardware para que a comparação seja mais indicativa do desempenho do software de servidor. (Poderá exigir acesso root.)

★★ **E27.5** No laboratório, usando um analisador de pacotes (**tcpdump**), monitore uma sessão telnet entre o Windows e o Linux. Obtenha e instale o software PuTTY e repita o monitoramento. Em cada caso, o que você pode observar com o analisador de pacotes? (Requer acesso root.)

★★ **E27.6** Configure o Samba para compartilhar arquivos com um PC Windows em sua rede local. Explore a idéia de exportar o seu diretório de usuário para um PC e armazenar seus arquivos de dados do PC no computador Linux. Um computador Linux rodando Samba é uma boa alternativa em relação a manter arquivos localmente num PC Windows? Há aplicativos Windows que requeiram que dados estejam armazenados no lado Microsoft?

Snow666

28 Daemons

Um daemon é um processo em segundo plano que realiza uma função específica ou tarefa relacionada ao sistema. Mantendo-se a filosofia de modularidade do UNIX e do Linux, os daemons são programas em vez de partes do kernel. Muitos daemons iniciam no momento da inicialização e continuam a ser executados enquanto o sistema estiver ligado. Outros daemons são iniciados quando preciso e executados apenas enquanto ainda forem úteis.

"Daemon" foi usado primeiramente como um termo computacional idealizado por Mick Bailey, um cidadão britânico que estava trabalhando na equipe de programação do CTSS no MIT durante o início dos anos 60.[1] Mick se baseou na citação do Oxford English Dictionary tanto em termos de significado quanto de pronúncia da palavra. As palavras "daemon" e "demon" provêm da mesma raiz, mas "daemon" é uma forma mais antiga e seu significado é um tanto diverso. Um daemon é um espírito acompanhante que influencia o caráter ou a personalidade de uma pessoa. Os daemons não são subordinados do mal *ou* do bem; eles são criaturas de pensamento e vontade independentes. Os daemons trilharam seu caminho do CTSS para o Multics, depois para o UNIX e finalmente para o Linux, no qual são tão populares que precisam de um superdaemon (**inetd** ou **xinetd**) para gerenciá-los.

Este capítulo apresenta uma visão geral dos daemons Linux mais comuns. Nem todos os daemons aqui listados são fornecidos com todas as distribuições Linux e nem todo daemon fornecido com algumas distribuições Linux é listado aqui. Além de deixá-lo mais ciente sobre o funcionamento do Linux, um conhecimento do que todos os diversos daemons fazem irá torná-lo realmente preparado para quando um de seus usuários perguntar: "O que **klogd** faz?".

Antes de **inetd** ter sido escrito, todos os daemons iniciavam no momento do boot e rodavam continuamente (ou mais precisamente, eles paravam aguardando novo trabalho). Ao longo do tempo, um número cada vez maior de daemons foi adicionado ao sistema. A população de daemons se tornou tão grande que começou a provocar problemas de desempenho. Em resposta, os gurus da Berkeley desenvolveram **inetd**, um daemon que é responsável por iniciar outros daemons à medida que forem necessários. **inetd** fez tamanho sucesso que todas as versões do UNIX e Linux agora o incluem e a maioria dos daemons novos é executada sob seu controle.

Há vários daemons com os quais os administradores de sistemas devem estar intimamente familiarizados, seja porque eles requerem uma grande dose de administração seja por desempenharem um grande papel na operação diária do sistema. Alguns daemons descritos aqui em uma ou duas linhas possuem um capítulo inteiro dedicado a eles neste livro. Fornecemos referências cruzadas onde apropriado.

---

1. Esse pequeno histórico vem de Jerry Saltzer no MIT, via Dennis Ritchie.

www.snow666.blogspot.com
Snow666

Começamos este capítulo introduzindo uma série de daemons de sistema importantes (**init** e **cron**) e depois avançamos para uma discussão de **inetd** e **xinetd**. Finalmente, descrevemos brevemente a maioria dos daemons com os quais provavelmente o administrador de sistemas irá se deparar nas três distribuições adotadas neste livro.

28.1 INIT: O PROCESSO PRIMORDIAL

init é o primeiro processo a ser executado depois de o sistema ser inicializado, e sob diversos aspectos ele é o daemon mais importante. Ele sempre possui um PID igual a 1 e é um ancestral de todos os processos de usuário e de todos, exceto alguns poucos, os processos de sistema.

Na inicialização, **init** coloca o sistema num modo monousuário ou começa a executar os scripts necessários para trazer o sistema para o modo multiusuário. Ao inicializar o sistema no modo monousuário, **init** executa os scripts de inicialização depois de o shell monousuário ter sido terminado digitando-se **exit** ou <Control-D>.

Veja o Capítulo 7 para obter maiores informações sobre TTYs.

No modo multiusuário, **init** é responsável pela certificação de que os processos se encontram disponíveis para manipular logins em cada dispositivo habilitado por login. Logins em portas seriais são geralmente manipulados por alguma variante de **getty** (por exemplo, **agetty**, **mgetty** ou **mingetty**; consulte a página 140 para obter detalhes). Em muitos sistemas, **init** também supervisiona um procedimento de login gráfico que permite aos usuários logarem diretamente no X Windows.

Além de suas tarefas de gerenciamento de login, **init** também possui uma tarefa macabra de exorcizar processos zumbis que poderiam de outra maneira acumularem-se no sistema. O papel de **init** nesse processo é descrito na seção 5.4.

*Consulte a seção 7.9 para obter maiores informações sobre o arquivo **inittab**.*

init define vários "níveis de execução" que determinam que conjunto de recursos de sistema deve ser habilitado. Há sete níveis numerados de 0 a 6. O nome "s" é reconhecido como um sinônimo para nível 1 (modo monousuário). As características de cada nível de execução são definidas no arquivo **/etc/inittab**.

Normalmente é passado a **init** seu nível de execução na forma de um argumento por meio do carregador de inicialização. Se for especificado "s", **init** entrará no modo monousuário. Caso contrário, ele varrerá **/etc/inittab** em busca de entradas que se aplicam ao nível de execução solicitado e executa seus comandos correspondentes.

O comando **telinit** altera o nível de execução de **init** assim que o sistema estiver ativo. Por exemplo, **telinit 4** força **init** a ir para o nível de execução 4. O argumento mais útil de **telinit** é **-q**, que faz com que **init** releia o arquivo **/etc/inittab**.

As distribuições Linux geralmente implementam uma camada adicional de abstração sobre o mecanismo básico de nível de execução fornecido por **init**. Essa camada extra permite que pacotes de software individuais instalem seus próprios scripts de inicialização sem modificar o arquivo genérico **inittab** do sistema. Trazer **init** para um novo nível de execução faz com que os scripts apropriados sejam executados com os argumentos **start** ou **stop**.

Uma discussão mais completa sobre **init** e scripts de inicialização começa na página 22.

28.2 CRON E ATD: COMANDOS DE AGENDAMENTO

O daemon **cron** (conhecido como **crond** no Red Hat) é responsável pela execução de comandos em momentos predeterminados. Ele aceita arquivos de agendamento ("crontabs") tanto de usuários quanto de administradores.

cron é empregado freqüentemente para fins administrativos, inclusive gerenciamento de arquivos de log e limpeza do sistema de arquivos. De fato, **cron** é tão importante para os administradores de sistema que foi dedicado um capítulo inteiro a ele – o Capítulo 9, *Processos Periódicos*.

O daemon **atd** executa comandos agendados através do comando **at**.

28.3 INETD E XINETD: DAEMONS DE GERENCIAMENTO

inetd é um daemon que controla outros daemons. Ele inicia os seus daemons-cliente quando houver trabalho para eles e permite que se extingüiam graciosamente assim que suas tarefas tiverem sido completadas.

inetd funciona somente com daemons que fornecem serviços pela rede. Para saber quando alguém está tentando acessar um de seus clientes, **inetd** se autoconecta às portas de rede que normalmente seriam controladas pelos daemons quiescentes. Quando ocorre uma conexão, **inetd** inicia o daemon apropriado e conecta seus canais de I/O padrão à porta da rede. Os daemons têm de ser escritos tendo-se essa convenção em mente para serem compatíveis com **inetd**.

*Consulte a seção 21.8 para obter maiores informações sobre **tcpd**.*

Algumas distribuições Linux substituíram o **inetd** tradicional pelo **xinetd** de Panos Tsirigotis, uma alternativa "envenenada" que incorpora recursos de segurança similares àqueles anteriormente conseguidos através do uso de **tcpd**, o pacote de "envoltórios

www.snow666.blogspot.com
Snow666

TCP". **xinetd** também fornece uma melhor proteção contra ataques de recusa de serviços, melhores recursos de gerenciamento de logs e uma linguagem de configuração mais flexível. Infelizmente, a configuração de **inetd** não é compatível com versões mais novas em relação ao **xinetd**. Discutimos primeiramente o **inetd** tradicional e depois damos uma olhada em **xinetd** numa seção separada.

Entre os sistemas Linux adotados como exemplos neste livro, o SuSE e o Debian usam o **inetd** padrão, e o Red Hat tem como padrão o **xinetd**. Você pode converter qualquer sistema para usar o gerenciador de daemons não padrão, porém realmente não há nenhuma razão que o leve a fazer isto.

Alguns daemons (como aqueles associados ao NIS e NFS) dependem de uma camada adicional de indireção conhecida como sistema RPC (Remote Procedure Call, chamada procedural remota). O RPC foi originalmente desenvolvido e implementado pela Sun como uma maneira de promover o compartilhamento de informações num ambiente de rede heterogêneo. As atribuições de portas para daemons que usam RPC são controladas pelo daemon **portmap**, discutido posteriormente neste capítulo. Alguns daemons podem ser usados tanto da maneira tradicional (na qual são inicializados uma só vez e continuam a ser executados até que o sistema seja desligado) ou via **inetd**. Os daemons discutidos neste capítulo são marcados com um T caso sejam compatíveis com **inetd**.

Configurando inetd

inetd consulta **/etc/inetd.conf** para determinar quais portas da rede ele deve escutar. Eis um exemplo (sintetizado) de um sistema SuSE:

```
# Modelo de arquivo /etc/inetd.conf – de um sistema SuSE
#
time      stream tcp nowait     root internal
time      dgram  udp wait       root internal
ftp       stream tcp nowait     root /usr/sbin/tcpd in.ftpd
# telnet  stream tcp nowait     root /usr/sbin/tcpd in.telnetd
# login   stream tcp nowait     root /usr/sbin/tcpd in.rlogind
# exec    stream tcp nowait     root /usr/sbin/tcpd in.rexecd
pop3      stream tcp nowait     root /usr/sbin/tcpd /usr/sbin/popper -s
# imap    stream tcp nowait     root /usr/sbin/tcpd imapd
# finger  stream tcp nowait nobody/usr/sbin/tcpd in.fingerd -w
# swat é o Samba Web Administration Tool
swat      stream tcp nowait.400 root /usr/sbin/swat swat
rsync     stream tcp nowait     root /usr/sbin/tcpd /sbin/rsyncd --daemon
...
```

A primeira coluna contém o nome do serviço. **inetd** associa nomes de serviços a números de portas consultando o arquivo **/etc/services** (para serviços TCP e UDP) ou então o arquivo **/etc/rpc** e o daemon **portmap** (para serviços RPC). Os serviços RPC são identificados por nomes no formato *nome/num* (onde *num* é o número da versão do protocolo) e *nome* a designação `rpc` na coluna três. O único serviço RPC que é comumente controlado por **inetd** é **mountd**, o daemon de montagem NFS. As distribuições Linux parecem executar esse daemon da maneira antiga (iniciando-o no momento da inicialização), de forma que talvez você não tenha serviços RPC nenhum em seu arquivo **inetd.conf**.

Num host com mais de uma interface de rede, você pode anteceder o nome do serviço com uma lista de endereços IP separados por vírgulas ou nomes de host simbólicos para indicar as interfaces nas quais **inetd** deve escutar solicitações de serviços. Por exemplo, a linha

```
inura:time stream tcp nowait root internal
```

fornece o serviço time somente na interface associada ao nome inura em DNS, NIS ou o arquivo **/etc/hosts**.

A segunda coluna determina o tipo de soquete que o serviço usará e é invariavelmente `stream` ou `dgram`. `stream` é usado com serviços TCP (orientados a conexões) e `dgram` é usado com UDP.

A terceira coluna identifica o protocolo de comunicação usado pelo serviço. Os tipos admissíveis são listados no arquivo **protocols** (usualmente em **/etc**). O protocolo é quase sempre `tcp` ou `udp`. Serviços RPC anexam antes `rpc/` ao tipo de protocolo.

Se o serviço que está sendo descrito é capaz de processar várias solicitações por vez (em vez de processar uma solicitação e sair), a coluna quatro deve ser configurada em `wait`. Essa opção permite que o daemon gerado assuma o controle da porta enquanto ele estiver em execução; **inetd** aguarda o daemon sair antes de retomar seu monitoramento da porta. O contrário de `wait` é `nowait`; ele faz com que **inetd** monitore continuamente e crie uma cópia do daemon cada vez que ele receber uma solicitação. A seleção de `wait` ou `nowait` tem de corresponder ao verdadeiro comportamento do daemon e não deve ser configurada arbitrariamente. Ao configurar um novo daemon, consulte o

Snow666

arquivo **inetd.conf** para ter um exemplo de linha de configuração ou consulte as páginas de manual para o daemon em questão.

A forma `nowait.400`, usada na linha de configuração para **swat**, significa que **inetd** deve gerar no máximo 400 instâncias do daemon de servidor por minuto. O padrão é mais conservador, 40 instâncias por minuto. Dada a natureza deste serviço (uma ferramenta administrativa para o Samba), não fica claro porque o limiar regulador foi aumentado.

A quinta coluna dá o nome de usuário sob o qual o daemon deve ser executado. Se você não confiar num determinado programa ou souber de antemão que ele apresenta problemas de segurança, poderá executá-lo como qualquer outro usuário que não seja o root para reduzir o nível de exposição. Obviamente, essa técnica funciona somente para daemons que não requeiram poderes de root. No exemplo anterior, **in.fingerd** é executado como o usuário "nobody".

Os campos remanescentes dão o nome de caminho totalmente qualificado do daemon e seus argumentos de linha de comando. A palavra-chave `internal` indica serviços cujas implementações são fornecidas pelo próprio **inetd**.

*Consulte a seção 21.8 para obter maiores informações sobre **tcpd**.*

A maioria das entradas de serviço nesse exemplo executa seus daemons via **tcpd** em vez de executá-los diretamente. **tcpd** registra em log tentativas de conexão e implementa controle de acesso baseado na origem da tentativa de conexão. Em geral, todos os serviços devem ser protegidos com **tcpd**. Esse modelo de configuração apresenta um problema de segurança em potencial já que **swat**, um utilitário de configuração para compartilhamento de arquivos, não é protegido.[2]

Pelo fato de **inetd** ser responsável pelo gerenciamento de vários serviços comuns baseados em rede, ele desempenha um papel importante em assegurar seu sistema. É importante verificar que somente serviços que você realmente precisa e confia foram habilitados em **inetd.conf**. Num sistema novo, é quase certo que você precisará modificar **inetd.conf** para desabilitar serviços desnecessários ou indesejáveis em seu ambiente. Uma excelente regra é habilitar somente os serviços absolutamente necessários e desativar todos os demais.

Note que nesse exemplo, os servidores para **rlogin**, **telnet**, **finger** e **rexec** foram desabilitados. Recomendamos que esses serviços sejam desabilitados em todos os sistemas de produção. Veja a seção 21.7, *Problemas de Segurança Variados*, para uma explicação mais detalhada.

Veja o Capítulo 11 para maiores informações sobre syslog.

Mudanças em **/etc/inetd.conf** não entram em vigor até que você diga a **inetd** para lê-lo novamente, o que é feito enviando-se um sinal de hangup para **inetd**. Depois de feita a sinalização, aguarde um minuto e verifique os arquivos de log em busca de mensagens de erro relacionadas às suas alterações (**inetd** registra os erros em syslog sob o recurso "daemon"). Teste qualquer serviço novo que você tenha adicionado para garantir que funcione corretamente.

Configurando xinetd

Algumas distribuições fornecem tanto **inetd** quanto **xinetd**, embora somente um está habilitado como opção-padrão. Use aquele com o qual você se sentir mais à vontade. **xinetd** é padrão em sistemas Red Hat.

O principal arquivo de configuração de **xinetd** é **/etc/xinetd.conf**. Ele inclui grande parte das mesmas informações do arquivo **inetd.conf**, mas usa um formato de lista de atributo/valor (em vez de formato tabular). A seguir apresentamos um exemplo que define alguns padrões de sistema e um serviço FTP, depois incluímos os arquivos no diretório **/etc/xinetd.d** (como no Red Hat). Pacotes individuais podem colocar arquivos de configuração no formato **xinetd.conf** no diretório **xinetd.d** e fazer com que eles sejam coletados automaticamente pelo sistema — um recurso elegante.

```
defaults
{
        log_type        = SYSLOG local4 info
        log_on_success  = HOST EXIT
        log_on_failure  = HOST ATTEMPT RECORD
        instances       = 2
}
service ftp
{
        # Instâncias ilimitadas pois wu.ftpd realiza o seu próprio controle de carregamento
        socket_type     = stream
        protocol        = tcp
        wait            = no
```

2. Se estiver tentando usar **tcpd** para proteger um serviço, o primeiro argumento de linha de comando do daemon sempre deve ser o nome reduzido do próprio daemon. Essa exigência não é uma peculiaridade de **inetd**, mas uma convenção tradicional do UNIX que normalmente é oculta pelo shell.

Snow666


```
        user            = root
        server          = /usr/sbin/wu.ftpd
        server_args     = -a
        instances       = UNLIMITED
        only_from       = 128.138.0.0/16
        log_on_success += DURATION
}

includedir /etc/xinetd.d
...
```

A Tabela 28.1 fornece um miniglossário de parâmetros.

Alguns parâmetros de **xinetd** podem aceitar atribuições na forma `+=` ou `-=` (conforme visto no valor `log_on_success` para o servidor FTP) para modificar os valores-padrão em vez de substituí-los totalmente. Em termos genéricos, essa sintaxe é permitida onde ela fizer sentido. Somente alguns parâmetros são realmente necessários para cada serviço; eles correspondem aos mesmos trechos de informação encontrados no arquivo **inetd.conf**.

Podem ser especificadas listas de concordância de endereços para os parâmetros `only_from` e `no_access` em diversos formatos. Os mais úteis são os endereços IP no formato CIDR com uma máscara explícita (conforme mostrado no exemplo) e nomes de host ou domínios como boulder.colorado.edu e .colorado.edu — observe o ponto precedente. Várias especificações podem ser separadas com um espaço (como em todas as listas **xinetd**).

xinetd pode tanto registrar diretamente num arquivo de log como submeter entradas de log para syslog. Já que o volume de informações de log pode ser potencialmente elevado num servidor atarefado, pode ser que faça sentido usar registro direto em arquivo por questões de desempenho. Tenha em mente que registrar informações de log num arquivo é menos seguro do que enviá-las para um servidor remoto através de syslog, pois um hacker que ganhe acesso ao sistema local pode adulterar os arquivos de log.

xinetd é capaz de fornecer alguns serviços interessantes como encaminhamento de solicitações a um host interno que não é visível ao mundo externo. Vale a pena revisar a página de manual referente a **xinetd** para ter uma idéia de seus recursos.

Tabela 28.1 Parâmetros de configuração de xinetd (não é uma lista completa).

Parâmetro	Valor	Significado
`bind`	*ipaddr/host*	Interface na qual se deve disponibilizar este serviço
`cps`	*num*	Limita o número de conexões por segundo
`disable`	*yes/no*	Desabilita serviço; mais fácil do que comentá-lo
`include`	*path*	Lê o caminho listado como um arquivo de configuração suplementar
`includedir`	*path*	Lê todos os arquivos no diretório especificado
`instances`	*num/UNLIMITED*	Número máximo de instâncias simultâneas
`log_on_failure`	*special*	Informações para log de falhas ou recursos de acesso[a]
`log_on_success`	*special*	Informações para log de conexões bem-sucedidas[a]
`log_type`	*special*	Configura arquivo de log ou parâmetros de syslog
`max_load`	*num*	Desabilita serviço caso a carga média seja > limiar
`nice`	*num*	Valor **nice** dos processos de servidor gerados
`no_access`	*matchlist*	Nega serviço a endereços IP especificados
`only_from`	*matchlist*	Aceita solicitações somente de endereços especificados
`per_source`	*num*	Limita o número de instâncias por máquina remota
`protocol`	*tcp/udp*	Protocolo de serviços
`server`	*path*	Caminho para o binário de servidor
`server_args`	*string*	Argumentos de linha de comando para o servidor[b]
`socket_type`	*stream/dgram*	Usa stream para serviços TCP, dgram para UDP
`user`	*username*	Usuário (UID) como qual o serviço deve ser executado
`wait`	*yes/no*	**xinetd** deve parar de se intrometer até que o daemon seja encerrado?

a. Note que a diretiva USERID usada com estes parâmetros fará com que xinetd realize consultas IDENT nas conexões, normalmente resultando em atrasos significativos.

b. Diferentemente de **inetd**, **xinetd** não requer que o comando de servidor seja o primeiro argumento.

Snow666

O arquivo services

Após adicionar um novo serviço a **inetd.conf** ou **xinetd.conf**, talvez você também precise criar uma entrada para ele no arquivo **/etc/services**. Esse arquivo é usado por várias rotinas de biblioteca padrão que fazem a associação entre nomes de serviços e números de portas. **xinetd** na verdade permite que você especifique diretamente o número da porta, mas sempre é uma boa idéia manter uma lista mestra das portas no arquivo **services**.

Por exemplo, ao digitar o comando

```
% telnet anchor smtp
```

telnet procura o número da porta para o serviço "smtp" no arquivo **services**. A maioria dos sistemas vêm com os serviços comuns já configurados; você somente precisa editar o arquivo **services** caso acrescente algo novo.

O arquivo **services** é usado somente para serviços TCP/IP confiáveis; informações similares para serviços RPC são armazenadas em **/etc/rpc**.

Eis algumas linhas selecionadas de um arquivo **services** (o original tem cerca de 70 linhas de comprimento[3]):

```
tcpmux      1/tcp                # multiplexador de portas TCP
echo        7/tcp
echo        7/udp
...
smtp        25/tcp   mail
time        37/tcp   timserver
time        37/udp   timserver
rlp         39/udp   resource    # localização de recursos
name        42/tcp               # IEN 116
whois       43/tcp   nicname
...
```

O formato de uma linha é

```
nome        porta/proto nomes-alternativos        # comentário
```

Os serviços são geralmente listados em ordem numérica, embora esta ordem não seja exigida. *nome* é o nome simbólico do serviço (o nome que você usa no arquivo **inetd.conf**). *porta* é o número da porta à qual o serviço normalmente escuta; se o serviço for controlado por **inetd**, é a porta que **inetd** escutará.[4]

proto indica o protocolo usado pelo serviço; na prática, é sempre `tcp` ou `udp`. Se um serviço puder usar tanto UDP quanto TCP, terá de ser incluída uma linha para cada (como acontece com o serviço `time` acima). O campo *nomes-alternativos* contém nomes adicionais para o serviço (por exemplo, `whois` também pode ser pesquisado como `nicname`).

portmap/rpcbind: associa serviços RPC a portas TCP e UDP

portmap associa números de serviços RPC a portas TCP/IP nas quais seus servidores estão escutando. Quando um servidor RPC é inicializado, ele se registra com **portmap**, listando os serviços que suporta e a porta na qual pode ser contactado. Clientes consultam **portmap** para descobrir como manter contato com o servidor apropriado.

Este sistema permite que uma porta seja associada a um nome de serviço simbólico. Ele é basicamente mais um nível de abstração acima do arquivo **services**, muito embora um que introduza complexidade adicional (e problemas de segurança) sem solucionar quaisquer problemas práticos.

Se o daemon **portmap** morrer, todos os serviços que dependem dele (inclusive **inetd** e NFS) terão de ser reinicializados. Em termos práticos, isso significa que é chegado o momento de reinicializar o sistema. **portmap** tem de ser iniciado antes de **inetd** para **inetd** manipular serviços RPC corretamente.

28.4 Daemons de Kernel

Por razões arquitetônicas, algumas partes do kernel do Linux são controladas como se fossem processos de usuário. Esses processos normalmente podem ser identificados pelos seus PIDs baixos e nomes que começam com a letra **k**, como **kflushd**, **kupdate**, **kpiod** e **kswapd** nos kernels 2.2 e **keventd**, **kapm-idled**, **kreclaimd**, **kupdated** no kernel 2.4. Em grande parte, esses processos lidam com vários aspectos do gerenciamento de memória e sincronização

3. Um tamanho aproximado para o arquivo **services** é de 360 linhas atualmente (N.R.T.).
4. Os números das portas não são arbitrários. Todas as máquinas têm de concordar sobre quais serviços se associam a quais portas; caso contrário, as solicitações serão constantemente direcionadas para a porta errada. Se você estiver criando um serviço específico a um site, selecione um número de porta alto (maior do que 1023) que já não se encontre listado no arquivo **services**.

Snow666

do cache de disco. Eles não podem ser manipulados pelo administrador de sistemas e geralmente devem ser deixados de lado.

Um outro daemon de sistema nessa categoria, muito embora um com um nome não padrão, é **mdrecoveryd**. Ele faz parte da implementação "dispositivos múltiplos", mais comumente conhecida como RAID.

klogd: lê mensagens de kernel

klogd é responsável por ler entradas de log do buffer de mensagens do kernel e por encaminhá-las para o syslog de modo que possam ser direcionadas aos seus destinos finais. Veja *Registro do horário de inicialização e do kernel* na seção 11.2 para maiores informações.

28.5 Daemons de Serviços de Arquivos

Os daemons a seguir fazem parte do sistema de compartilhamento de arquivos NFS. Aqui apresentamos apenas uma breve descrição de suas funções; o Capítulo 17 descreve-os em detalhes.

rpc.nfsd: serve arquivos

rpc.nfsd é executado em servidores de arquivos e manipula solicitações de clientes NFS. Na maioria das implementações do NFS, **nfsd** é realmente apenas uma parte do kernel que foi embelezada como um processo por motivos de escalonamento. O Linux na verdade ostenta duas implementações diferentes, uma das quais segue essa convenção e outra que é executada no espaço de usuário. A implementação no kernel é mais popular e o padrão para a maioria das distribuições.

rpc.nfsd aceita um único argumento que especifica quantas cópias de si próprio ele deve criar. Há uma certa magia negra envolvida na escolha do número correto de cópias; consulte a página 356.

rpc.mountd: responde a solicitações de montagem

rpc.mountd aceita solicitações de montagem de sistemas de arquivos de clientes NFS em potencial. Ele verifica se cada cliente tem permissão para montar os diretórios solicitados. **rpc.mountd** consulta o arquivo **/var/state/nfs/xtab** para determinar quais solicitantes são legítimos.

amd e automount: monta sistemas de arquivos por demanda

amd e **automount** são automontadores NFS, daemons que aguardam até que um processo tente usar um sistema de arquivos antes deles realmente montá-los. Os automontadores posteriormente desmontam os sistemas de arquivos caso não tenham sido acessados num período especificado.

O emprego de automontadores é muito útil em ambientes grandes em que dezenas ou centenas de sistemas de arquivos são compartilhados na rede. Os automontadores aumentam a estabilidade da rede e reduzem a complexidade da configuração, já que todos os sistemas da rede podem compartilhar a mesma configuração **amd** ou **automountd**. Cobrimos em detalhes o emprego destes daemons a partir da página 359.

rpc.lockd e rpc.statd: controla bloqueios NFS

Embora estes dois daemons sejam distintos, eles sempre são executados em equipe. **rpc.lockd** mantém bloqueios de alerta (como **flock** e **lockf**) em arquivos NFS. **rpc.statd** permite que processos monitorem o estado de outras máquinas que estão executando NFS. **rpc.lockd** usa **rpc.statd** para decidir quando tentar se comunicar com uma máquina remota.

rpciod: coloca blocos NFS em cache

rpciod coloca em cache solicitações de leitura e de gravação em clientes NFS. Ele executa tanto buffering antes da leitura quanto após gravação e melhora muito o desempenho do NFS. Este daemon é análogo aos daemons **biod** e **nfsiod** encontrados em outros sistemas, embora seja um tanto diferente em termos estruturais.

rpc.rquotad: serve cotas remotas

rpc.rquotad permite que usuários remotos verifiquem suas cotas em sistemas de arquivos que eles montaram com NFS. A verdadeira implementação das restrições de cotas ainda é realizada no servidor; **rpc.rquotad** simplesmente faz com que o comando **quota** funcione corretamente.

smbd: fornece serviços de arquivos e de impressão a clientes Windows

smbd é o servidor de arquivos e de impressão na família Samba. Ele fornece serviços de compartilhamento de arquivos e de impressão através do protocolo Windows conhecido de forma variada como SMB ou CIFS. Veja a página 595 para maiores detalhes.

Snow666

nmbd: o servidor de nomes NetBIOS

nmbd é mais um componente do Samba. Ele responde a solicitações de serviços de nomes NetBIOS geradas por máquinas Windows. Ele também implementa o protocolo de navegação que máquinas Windows usam para preencher a pasta Ambiente de Rede e torna visíveis os discos compartilhados no host local. **nmbd** também pode ser usado como servidor WINS.

28.6 Daemons de Bancos de Dados Administrativos

Vários daemons estão associados ao sistema de bancos de dados administrativos NIS da Sun, que é descrito no Capítulo 18, *Compartilhando Arquivos de Sistemas*. Embora o NIS tenha se originado na Sun, agora ele também é usado em sistemas de muitos outros fabricantes, inclusive o Linux.

ypbind: localiza servidores NIS

O daemon **ypbind** roda em todos os clientes e servidores NIS. Ele localiza um servidor NIS para o qual consultas podem ser direcionadas. **ypbind** não processa realmente as solicitações em si; ele simplesmente informa aos programas-cliente que servidor usar.

ypserv: servidor NIS

ypserv roda em todos os servidores NIS. **ypserv** aceita consultas de clientes e responde com as informações solicitadas. Consulte a partir da página 375 para maiores informações de como configurar as máquinas que executam **ypserv**.

rpc.ypxfrd: transfere bancos de dados NIS

rpc.ypxfrd transfere bancos de dados NIS para servidores-escravo de uma maneira eficiente. Um escravo inicia uma transferência com o comando **ypxfr**. Toda vez que um banco de dados for alterado no mestre, ele deve ser imediatamente transferido para todos os escravos de forma que os servidores NIS permaneçam consistentes entre si.

nscd: atribuir nome ao daemon de cache de serviços

nscd coloca em cache os resultados de chamadas das rotinas de biblioteca C padrão nas famílias **getpw***, **getgr*** e **gethost***, que pesquisam dados tradicionalmente armazenados nos arquivos **passwd**, **group** e **hosts**. Nos dias de hoje, a gama de fontes potenciais é maior e inclui opções como NIS e DNS. **nscd** realmente não sabe de onde os dados provêm; ele simplesmente coloca os resultados em cache e os usa para curto-circuitar chamadas de bibliotecas subseqüentes. A política de caching é configurada no arquivo **/etc/nscd.conf**.

28.7 Daemons Internet

Definimos os "daemons Internet" de maneira muito livre para representar daemons que usam protocolos Internet para manipular solicitações. Muitos daemons Internet na verdade gastam a maior parte do tempo atendendo a solicitações locais.

talkd: conecta a serviço de chat de rede

As solicitações de conexão do programa **talk** são manipuladas por **talkd**. Ao receber uma solicitação, **talkd** negocia com a outra máquina para estabelecer uma conexão de rede entre os dois usuários que executaram **talk**.

sendmail: transporta e-mails

Entre as tarefas de **sendmail** temos: aceitar mensagens de usuários e sites remotos, regravar endereços, expandir nomes alternativos e transferir e-mails através da Internet. **sendmail** é um daemon importante e muito complexo. Consulte o Capítulo 19, *E-Mail*, para informações completas.

snmpd: fornece serviço de gerenciamento de rede

snmpd responde a solicitações que usam o protocolo SNMP (Simple Network Management Protocol, protocolo de gerenciamento de rede simples). O SNMP padroniza algumas operações de rede comuns.
Veja a página 479 para obter maiores informações sobre o SNMP.

rwhod: mantém a lista de usuários remotos

rwhod é um remanescente dos "primórdios" (anos 80) e mantém informações sobre os usuários logados nas máquinas da rede. **rwhod** coleta essas informações para a máquina local e as transmite; ao receber informações de outros hosts, ele verifica se são razoáveis e depois as coloca no arquivo **/var/spool/rwho/whod.***nomedohost*,

Snow666

onde *nomedohost* é o nome do host que envia as informações. Os programas **rwho** e **ruptime** referem-se a esses arquivos.

Como padrão, **rwhod** transmite a cada três minutos, de modo que as informações reportadas por **rwho** e **ruptime** são apenas aproximadamente corretas. **rwhod** é muito ineficiente, de maneira que, a menos que você tenha largura de banda de rede para queimar ou realmente use estas informações, deve desativá-lo.

ftpd: servidor de transferência de arquivos

ftpd é o daemon que manipula solicitações de **ftp**, o programa de transferência de arquivos da Internet.

*Veja a seção 22.6 para maiores informações sobre **ftpd**.*

Muitos sites o desabilitam, seja devido a ser um voraz consumidor de recursos seja por estarem preocupados em relação à segurança do sistema. **ftpd** pode ser configurado para permitir que qualquer um transfira arquivos de e para sua máquina.

Uma série de implementações **ftpd** se encontram disponíveis para sistemas Linux. Caso planeje rodar um servidor com tráfego intenso ou precise de recursos avançados como gerenciamento de carga, pode ser sensato investigar as alternativas para o **ftpd** padrão de sua distribuição.

O WU-FTPD, desenvolvido na Universidade de Washington, é uma das alternativas mais populares ao padrão **ftpd**. Vá ao site www.wu-ftpd.org para maiores informações.

popper: servidor de caixa postal básico

O daemon **popper** implementa o POP (Post Office Protocol). Esse protocolo é comumente usado por sistemas não-Linux para receber e-mails.

imapd: servidor de caixa postal avançado

O daemon **imapd** implementa o IMAP (Internet Mail Access Protocol[5], protocolo de acesso a e-mails na Internet), que é uma alternativa mais festiva e cheia de recursos do que o POP. Ele permite que usuários baseados em PC (ou usuários Linux com leitores de e-mail preparados para o padrão IMAP) acessem seus e-mails de uma série de localidades, com pastas de e-mail armazenadas no servidor Linux. Vá ao endereço www.imap.org para obter maiores informações sobre o IMAP.

in.rlogind: servidor com login remoto

in.rlogind manipula logins remotos. Ao ser chamado por **inetd**, ele tenta automaticamente autenticar o usuário remoto examinando **/etc/hosts.equiv** e o arquivo **~/.rhosts** do usuário. Se a autenticação automática for bem-sucedida, o usuário é logado diretamente. Caso contrário, **in.rlogind** executa o programa **login** para solicitar ao usuário uma senha. Devido à sua fácil e barata autenticação, **in.rlogind** é um tanto preocupante em termos de segurança. Veja a página 494 para maiores comentários sobre o assunto.

in.telnetd: mais um servidor com login remoto

in.telnetd é similar a **in.rlogind**, exceto pelo fato de usar o protocolo TELNET. Esse protocolo permite que ambos os lados (cliente e servidor) negociem configurações duplex e controle de fluxo, tornando-o uma melhor opção do que **in.rlogind** para links que são lentos ou não confiáveis.

Da mesma forma que **rlogin**, **telnet** transmite senhas em texto puro através da rede. Portanto, o seu emprego é desencorajado em redes modernas. Porém, muitos outros sistemas não-Linux suportam **telnet**.

sshd: servidor com login remoto seguro

sshd fornece serviços similares a **in.rlogind**, mas suas sessões são transportadas (e autenticadas) através de um duto criptografado. Uma série de algoritmos criptográficos estão disponíveis. Devido ao ambiente hostil da Internet dos dias de hoje, você tem de permitir acesso proveniente da Internet *somente* através de um daemon como este — e *não* **in.rlogind** ou **in.telnetd**. Você poderá encontrar maiores informações sobre **sshd na seção** 21.9.

in.rshd: servidor de execução de comandos remoto

in.rshd manipula solicitações de execução de comandos remotos por meio de **rsh** e **rcmd**. O processo de autenticação aplicado por **in.rshd** é similar àquele de **in.rlogind**, exceto pelo fato de que se a autenticação automática não funcionar, **in.rshd** negará a solicitação sem permitir que o usuário forneça uma senha. **in.rshd** também é o servidor para **rcp** (**cp** remoto). Da mesma forma que **in.rlogind**, **in.rshd** tornou-se uma espécie de pária nos últimos anos por questões de segurança. Veja a seção 21.7 para maiores informações.

5. O nome correto é Internet Message Access Protocol, definido na RFC 3501 (N.R.T.).

Snow666

in.rexecd: mais um servidor para execução de comandos

in.rexecd é similar a **in.rshd**, exceto pelo fato de não realizar autenticação automática; todas as solicitações têm de ser acompanhadas por um nome de usuário e por uma senha. Este servidor foi usado por alguns programas de rede primitivos, mas ele não está mais em uso amplamente. Pelo fato do nome de usuário e senha serem transmitidos em texto puro através da rede, **in.rexecd** é um método extremamente pobre para controlar execução remota.

rsyncd: sincroniza arquivos entre vários hosts

rsyncd é realmente apenas um link ao comando **rsync**; a opção **--daemon** o transforma num processo de servidor. **rsyncd** facilita a sincronização de arquivos entre os hosts. Ele é, essencialmente, uma versão eficiente e atenta à segurança de **rcp**. **rsync** é um verdadeiro achado para os administradores de sistemas, e neste livro descrevemos seu uso numa série de contextos diferentes. Veja a seção 18.1 para informações genéricas e algumas dicas sobre como usar **rsync** para compartilhar arquivos de sistema. **rsync** também é um grande participante dos processos de instalação internos de muitos sites. Veja a seção 23.4 para maiores detalhes sobre essa aplicação.

routed: mantém tabelas de roteamento

routed mantém as informações de roteamento usadas por TCP/IP para enviar e encaminhar pacotes numa rede. **routed** lida somente com roteamento dinâmico; rotas que são definidas estaticamente (isto é, embutidas na tabela de roteamento do sistema com o comando **route**) jamais são modificadas por **routed**. **routed** é relativamente estúpido e ineficiente; recomendamos seu uso somente em algumas poucas situações específicas. Veja a seção 14.4 para uma discussão mais detalhada de **routed**.

gated: mantém tabelas de roteamento complicadas

gated compreende vários protocolos de roteamento, inclusive RIP, o protocolo usado por **routed**. **gated** transforma informações de roteamento entre os diversos protocolos e é muito maleável em termos de configuração. Ele também pode ser muito mais gentil com sua rede do que **routed**. Veja a seção 14.4 para obter maiores informações sobre **gated**.

named: servidor DNS

named é o servidor mais popular para DNS (Domain Name System, sistema de nomes de domínios). Ele transforma nomes de host em endereços de rede e realiza muitos outros recursos e truques, todos usando um banco de dados distribuído mantido por **named**s em toda parte. O Capítulo 16, *O Sistema de Nomes de Domínios*, descreve os cuidados e como alimentar **named**.

Veja a seção11.4 para obter maiores informações sobre syslog.

syslogd: processa mensagens de log

syslogd atua como uma alfândega para informações de estados e mensagens de erro geradas pelos daemons e software de sistema. Antes de **syslogd** ter sido desenvolvido, os daemons gravavam suas mensagens de erro diretamente na console do sistema ou mantinham seus próprios arquivos de log privados. Agora eles usam a rotina de biblioteca **syslog** para transferir as mensagens para **syslogd**, que as organiza de acordo com regras estabelecidas pelo administrador do sistema.

in.fingerd: procura usuários

in.fingerd fornece informações sobre os usuários que estão logados no sistema. Se solicitado, ele também pode fornecer mais detalhes sobre usuários individuais. **in.fingerd** realmente não faz muito por si só: ele simplesmente aceita linhas de entrada e as passa ao programa **finger** local.

finger é capaz de retornar uma quantidade razoável de informações sobre um usuário, inclusive o estado de login do usuário, o conteúdo do campo GECOS do usuário em **/etc/passwd** e o conteúdo dos arquivos de usuário **~/.plan** e **~/.project**.

Se estiver conectado à Internet e usando **in.fingerd**, qualquer pessoa no mundo poderá obter essas informações. **in.fingerd** realmente habilitou muitos serviços interessantes (como as listas telefônicas na Internet), mas também permitiu que pessoas implementassem uma série de esquemas fraudulentos, como localizar pessoas e ligar para elas para tentar vender coisas ou serviços e para localizar endereços candidatos a receber spam. Alguns sites responderam a essa invasão desativando **in.fingerd**, ao passo que outros simplesmente restringiram a quantidade de informações por ele retornada. Se optar por usar **in.fingerd**, você deverá instalar uma versão atual; uma brecha de segurança em **fingerd**s mais antigos foi usada pelo primeiro worm da Internet em 1988.

www.snow666.blogspot.com
Snow666

httpd: servidor World Wide Web

httpd permite que seu site se torne um servidor Web. **httpd** é capaz de enviar texto, imagens e som a seus clientes. Veja o Capítulo 22, *Hospedagem Web e Servidores Internet*, para maiores informações sobre como servir páginas Web.

Veja a seção 24.4 para maiores informações sobre ***lpd****.*

lpd: controla a impressão

lpd é responsável pelo sistema de spooling de impressão. Ele aceita tarefas de usuários e cria processos que realizam a impressão em si. **lpd** também é responsável pela transferência de tarefas de impressão de e para sistemas remotos. **lpd** por vezes pode travar e precisar ser reiniciado manualmente. Seu sistema pode ter a versão original do **lpd** ou a viçosa versão que faz parte do pacote LPRng. Veja o Capítulo 24, *Impressão*, para obter maiores informações sobre essas alternativas.

28.8 DAEMONS PARA SINCRONIZAÇÃO DE TEMPO

À medida que os computadores ficaram cada vez mais interdependentes, tornou-se cada vez mais importante para eles compartilhar uma idéia de tempo consistente. Relógios sincronizados são essenciais para correlacionar entradas de arquivos de log no caso de uma brecha de segurança e eles também são importantes para uma série de aplicativos de usuário final, desde projetos conjuntos de desenvolvimento de software até o processamento de transações financeiras.

timed: sincroniza relógios

Há diversos sistemas diferentes de sincronização de tempo, e mais de um daemon de tempo é chamado **timed**. A maioria dos sistemas usa, basicamente, o mesmo esquema. Uma ou mais máquinas são designadas como mestras de tempo. Seus relógios são considerados autorizados, e eles negociam entre si para chegarem a um acordo sobre o horário "correto". Outras máquinas são escravas; elas conversam periodicamente com um mestre para saberem o horário e depois ajustam seus relógios internos.

O tempo entre os ajustes do relógio de um escravo é curto o bastante para que sejam necessários somente ligeiros ajustes. Os escravos usam a chamada de sistema **adjtimex** para facilitar o ajuste do relógio do sistema e impedir grandes diferenças de tempo (a menor ou a maior).[6] É especialmente perigoso atrasar o relógio de repente; o tempo deve ser uma função de crescimento monotônico.

A noção de horário "correto" é definida de uma forma um tanto nebulosa. Alguns sistemas consultam a rede para calcular um horário médio, ao passo que outros declaram um mestre correto por "decreto".

ntpd e xntpd: sincroniza ainda melhor os relógios

ntpd e **xntpd** são daemons que usam o Network Time Protocol definido na RFC-1119 para sincronizar uma série de relógios de máquinas com uma diferença entre eles dentro da faixa de milissegundos. Os servidores são dispostos numa árvore hierárquica, sendo cada um dos níveis chamado de "estrato". Clientes podem acessar uma série de padrões de tempo de referência, como aqueles fornecidos pelo WWV e GPS. Conseqüentemente, o NTP fornece uma maneira muito mais acurada de acertar o relógio numa máquina do que é oferecida por **timed**; os relógios não são somente sincronizados como também são precisos dentro de uma faixa de poucos milissegundos.

28.9 DAEMONS DE INICIALIZAÇÃO E DE CONFIGURAÇÃO

Nos anos 80, o mundo UNIX era varrido por uma mania de estações de trabalho sem disco ("diskless"). Essas máquinas eram inicializadas completamente pela rede e realizavam todas as suas operações de disco por meio de uma tecnologia de sistemas de arquivos remotos como o NFS. À medida que os preços dos discos foram caindo e suas velocidades aumentando, o interesse por estações de trabalho sem disco desapareceu rapidamente. Entretanto, elas podem voltar à moda a qualquer momento, como os sapatos plataforma dos anos 70. Os dois principais remanescentes da era "diskless" são uma abundância enorme de daemons desenvolvidos para suportarem sistemas sem disco e a bizarra organização dos sistemas de arquivos da maioria dos fabricantes.

Embora as estações de trabalho sem disco não sejam mais comuns hoje em dia, seus protocolos de inicialização foram usurpados por outros dispositivos. A maioria das impressoras de rede e hubs de rede gerenciáveis faz a inicialização através de alguma combinação dos serviços listados nesta seção.

6. **adjtimex** aumenta a velocidade do relógio do sistema para gradualmente cair no alinhamento correto. Quando o horário do sistema concordar com o horário objetivo atual, o ganho é cancelado e o relógio anda normalmente.

Snow666

dhcpd: atribuição dinâmica de endereços

O DHCP (Dynamic Host Configuration Protocol, protocolo de configuração dinâmica de host) dá aos PCs, laptops e outras plataformas "móveis" informações sobre seus endereços IP, gateway-padrão e servidor de nomes no momento da inicialização. **dhcpd** é o daemon que implementa este serviço no Linux. Você pode encontrar mais informações sobre o DCHP na página 209. Uma forma mais elaborada do DHCP chamada PXE ajuda máquinas compatíveis a serem inicializadas pela rede sem a necessidade de um dispositivo de inicialização local; veja a seção 23.2 para maiores detalhes.

in.tftpd: servidor de transferência de arquivos trivial

in.tftpd implementa um protocolo de transferência de arquivos similar àquele do **ftpd**, mas muito, muito mais simples. Muitos sistemas sem disco usam TFTP para fazer o download de seus kernels em um servidor. **in.tftpd** não executa autenticação, mas ele normalmente é restrito a servir arquivos num único diretório (normalmente **/tftpboot**). Já que qualquer coisa colocada no diretório TFTP passa a ser acessível para toda a rede, o diretório deve conter somente arquivos de inicialização e não deve ser gravável por todos.

rpc.bootparamd: suporte avançado essencial a sistemas sem disco

rpc.bootparamd usa o arquivo **/etc/bootparams** para informar clientes sem disco onde encontrar seus sistemas de arquivos. O serviço **rpc.bootparamd** é normalmente usado por máquinas que obtêm seus endereços IP usando RARP e que utilizam NFS para montar seus sistemas de arquivos.

28.10 Exercícios

★ **E28.1** Determine quais daemons estão sendo executados em seu sistema Linux usando **ps**. Determine também quais daemons se encontram disponíveis para serem executados através de **inetd.conf** ou **xinetd.conf**. Combine as listas e descreva o que cada daemon faz, onde ele é inicializado, se várias cópias podem (ou realmente) são executadas ao mesmo tempo e quaisquer outros atributos que você consiga recolher de **inetd.conf** e da saída de **ps**.

★ **E28.2** No laboratório, habilite o daemon de rede time, **ntpd**.

a) Como você sabe se o seu sistema está com o horário correto?

b) Configure manualmente o seu horário de sistema para estar 15 segundos atrasado por meio do comando **date**. Quanto tempo leva (ou levará) para o horário se tornar correto?

c) Configure manualmente o horário de seu sistema um mês adiantado. Como **ntpd** responde a essa situação? (Requer acesso root.)

★★ **E28.3** No laboratório, use uma ferramenta como **netstat** para determinar quais portas estão num estado de "escuta" em sua máquina.

a) Como você pode reconciliar as informações de **netstat** com aquilo que é encontrado em **inetd.conf**? Se houver uma discrepância, o que mais está ocorrendo?

b) Instale a ferramenta **nmap** numa máquina diferente. Execute uma varredura de portas com o seu sistema como alvo para verificar o que você aprendeu no item a. Que informações adicionais (se realmente houver alguma) você obteria de **nmap** que não eram óbvias de **netstat**? (Veja a seção 21.8 para obter maiores informações sobre **nmap**.)

www.an0w666.blogspot.com
Sn0w666

29 Políticas e Diretrizes

Este capítulo cobre alguns tópicos não técnicos que normalmente são incluídos no repertório de um administrador de sistemas. Além de discutir os diversos aspectos legais e políticos, falamos de alguns dos aspectos interpessoais do administrador de sistemas e as intrigas políticas que ocorrem em instalações Linux.

O Linux e as redes de computadores são bem jovens — o Linux com cerca de 12 anos de idade[1] e as redes com talvez 25 —, embora eles sejam um microcosmo de questões sociais que existem há milhares de anos. Em muitos casos, as instituições legais e sociais têm sido lentas para se adaptarem às implicações da nova tecnologia.

A Internet está firme em sua trajetória de substituir ou de pelo menos alterar muitos dos grandes componentes da infra-estrutura econômica: os setores editorial, de telecomunicações e de entretenimento, o serviço postal e agentes e intermediários de toda estirpe e cor — agentes de viagem, vendedores de livros, lojas de disco, corretores e assim por diante. Em vários desses contextos, nossas capacidades tecnológicas e financeira parecem ter avançado muito mais do que a infra-estrutura de diretrizes que é necessária para suportá-los. Por exemplo, existem leis e convenções bem definidas em relação à privacidade, uso e abusos em relação às correspondências tradicionais (papel). Mas e quanto ao e-mail? Ele é privado ou o dono do disco no qual é armazenado tem o direito de lê-lo? Se um computador encaminha uma mensagem, o dono deste computador pode ser responsabilizado caso a mensagem seja difamatória ou obscena?

As linhas referentes à propriedade intelectual parecem ficar cada vez mais confusas. É bem estabelecido que você pode deixar alguém ouvir o seu CD novo ou até mesmo emprestar este CD a um amigo. E é legal você transferir a música para o seu computador e ouvi-la desta maneira. Mas o que acontece se você quiser reproduzir o CD via Internet de seu trabalho? O que acontece se o seu amigo também escutar o CD dessa maneira? O que acontece se apenas um por vez de vocês ouvir o CD?

Aplicativos como o Napster (napster.com) e todos os novos aplicativos de rede não hierarquizados que ele gerou colocaram ao extremo esse conceito de compartilhamento via Internet. O Napster chegou ao campus universitário no 3º trimestre de 1999 e instantaneamente dobrou as contas de serviços Internet de muitos sites. A indústria fonográfica processou a Napster por violação de direitos autorais e ganhou. O caso baseou-se no Digital Millennium Copyright Act of 1998, uma lei americana que muitos acham que reprime enormemente os direitos de liberdade de expressão que são fundamentais nos Estados Unidos.

Como era de esperar, dada a incerteza que envolve tais questões, muitos sites (sem mencionar os governos) não têm uma política bem definida para lidar com elas.

1. O kernel Linux foi lançado em 1991 por Linus Torvalds. Assim, em 2003, ele tinha 12 anos (N.R.T.).

www.snow666.blogspot.com
Snow666

29.1 A Cultura Linux

Nos anos 80 e 90 podíamos diferenciar usuários de PCs de usuários de Macs pelos seus níveis de intensidade. Os usuários de PCs descobriram que seus computadores eram uma ferramenta útil, mas os usuários de Mac amavam suas máquinas. O computador de um usuário de Mac fazia parte da família, como o animal de estimação preferido.

A mesma intensidade que permeava o mundo Mac agora é muito forte na comunidade Linux. Os usuários do Linux não apenas gostam de seus sistemas — eles estão preparados para batalharem por eles, consertá-los e torná-los melhores, mais rápidos e mais seguros do que um computador Windows jamais sonhou em ser. Energia está fluindo no Linux numa velocidade incrível, e a cultura Linux possui suas próprias éticas, mitos, deuses e heróis.

Um grupo de entusiastas do Linux do Bergen Linux User Group em Bergen, Noruega, estava intrigado por uma destas RFCs tipo Dia da Mentira do IETF: RFC1149, *A Standard for the Transmission of IP Datagrams on Avian Carriers*. Essa RFC define o CPIP (Carrier Pigeon Internet Protocol), que a equipe implementara usando um bando de pombos-correio de um vizinho. Eis um teste da implementação deles (extraído do site do projeto em blug.linux.no/rfc1149):

```
Script started on Sat Apr 28 11:24:09 2001
vegard@gyversalen:~$ /sbin/ifconfig tun0
tun0      Link encap:Point-to-Point Protocol
          inet addr:10.0.3.2 P-t-P:10.0.3.1 Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST MTU:150 Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0
          RX bytes:88 (88.0 b) TX bytes:168 (168.0 b)

vegard@gyversalen:~$ ping -i 900 10.0.3.1[2]
PING 10.0.3.1 (10.0.3.1): 56 data bytes
64 bytes from 10.0.3.1: icmp_seq=0 ttl=255 time=6165731.1 ms
64 bytes from 10.0.3.1: icmp_seq=4 ttl=255 time=3211900.8 ms
64 bytes from 10.0.3.1: icmp_seq=2 ttl=255 time=5124922.8 ms
64 bytes from 10.0.3.1: icmp_seq=1 ttl=255 time=6388671.9 ms

--- 10.0.3.1 ping statistics ---
9 packets transmitted, 4 packets received, 55% packet loss
round-trip min/avg/max = 3211900.8/5222806.6/6388671.9 ms
vegard@gyversalen:~$ exit

Script done on Sat Apr 28 14:14:28 2001
```

O Linux agita. Pode-se dizer que esses caras tiveram muito tempo nas mãos, porém é exatamente este tipo de criatividade e de entusiasmo que tornam o movimento de código-fonte aberto (e o fonte do Linux participa deste) tão poderoso e com tanta alegria.

Linux de Prateleira

Era comum ouvir-se frases como "Ah, não podemos usar Linux — ele não é suportado!" nos círculos corporativos. Mas o mundo tem se alterado de forma tão rápida que chega a assustar a Microsoft. O Linux está abrindo suas trilhas pelo mundo comercial. Ele normalmente chega com alguns universitários recém-contratados que haviam usado Linux na faculdade e que o utilizam seja de maneira aberta ou de maneira clandestina em suas máquinas desktop. Após estabelecerem um posto avançado, o Linux normalmente se torna a plataforma preferida para servidores de e-mail ou servidores Web, em que sua segurança, desempenho e expansibilidade o tornam preferível em relação a soluções proprietárias da Microsoft.

O Linux é o sistema operacional atual que cresce mais rapidamente. Ele pode ser usado em tudo, de mainframes a relógios de pulso. Em 2001, havia 2.300 aplicativos comerciais disponíveis para o Linux, 30% acima do ano anterior.

Alguns parceiros corporativos importantes agora estão distribuindo e suportando o desenvolvimento do Linux: IBM, Hewlett-Packard, Compaq, Silicon Graphics e Sun, para citar alguns. É interessante observar o casamento entre esses fabricantes enormes e pesados com uma cultura Linux dirigida por engenheiros de software jovens e inexperientes cuja energia normalmente compensa sua forma um tanto rude. A IBM teve de passar por uma mudança de

2. Observe o uso apropriado do espaço de endereços privado da RFC1918. Claramente, uma implementação NAT baseada em pombos-correio seria necessária para conectar a rede de teste com o mundo externo.

Snow666

paradigma para se aclimatar à vida no mundo anárquico do fonte aberto do Linux. Ela foi capaz de contribuir, portando suas imensas bibliotecas de código bem testado. Veja os sites a seguir para maiores informações:

http://ibm.com/linux
http://oss.software.ibm.com/developerworks/opensource/linux

A SGI e outros fabricantes contribuíram com o Linux criando o Linux Test Project, uma série de conjuntos de testes e métodos de testes de resistência que validam a confiabilidade, a robustez e a estabilidade do Linux. Pelo menos um fornecedor de sistemas de segurança contra invasão de computadores reconhece essa qualidade, cobrando cerca de 5% a 15% menos para assegurar servidores Web Linux do que para assegurar sistemas Microsoft equivalentes. Um artigo de agosto de 2001 da ZDNet fez os seguintes comentários.[3]

> "A corretora de seguros J.S. Wurzler Underwriting Managers começou a cobrar 15% a mais em prêmios de clientes que usam o software Internet Information Server, da Microsoft, que o worm Code Red infestou... Wurzler, que vem vendendo seguros contra hackers desde 1998, baseou sua decisão em mais de 400 análises de seguros feitos por essa firma ao longo dos últimos três anos. Wurzler descobriu que os administradores de sistema trabalhando com sistemas de fonte aberto tendem a ser melhores treinados e permanecem mais tempo com seus funcionários do que em empresas que usam o software Windows."

Para algumas empresas, o custo de propriedade também é importante. O Linux é essencialmente livre, para começar, e também sua manutenção é mais fácil graças à sua gerência via rede. O Microsoft XP Professional custa atualmente cerca de US$ 275/assento somente para o sistema operacional. Assim que aplicativos relevantes são acrescentados, o preço para uma grande empresa pode ser assustador. Por exemplo, o google.com (o melhor mecanismo de busca na Web, em nossa opinião) tem atualmente 12.000 computadores Linux e fará uma atualização para 27.000 (mais ou menos) ao longo do próximo ano. Isso perfaz um pouco menos de US$7,5 milhões apenas em software de sistema operacional básico. Caramba!

O Linux é usado por vários órgãos governamentais americanos dentro de organizações como Nasa, Nersc, NIH, NOAA e USGS. Ele também é usado no Fermilab, Los Alamos, Oak Ridge e Sandia National Laboratories. Os computadores mais poderosos hoje em dia ainda são sistemas grandes — mas clusters Beowulf (veja www.beowulf.org) de máquinas Linux estão pegando.

O Linux alcançou a maturidade.

Projetos Linux

Além do Linux Test Project mencionado, duas outras importantes iniciativas na comunidade Linux merecem referência aqui: o LSB (Linux Standard Base (LSB) e o LDP (Linux Documentation Project). O LSB (www.linuxbase.org) tenta solucionar as diferenças existentes entre diferentes distribuições do Linux sob o ponto de vista do programador de aplicativos. Falaremos um pouco mais a respeito do LSB na seção 29.13.

O LDP (www.linuxdoc.org) fornece documentação em vários níveis de especialização técnica, inclusive longos folhetos introdutórios chamados guias, HOWTOs específicas para várias tarefas comuns, FAQs sobre diversos tópicos e páginas de manual para todos os pacotes e comandos principais do Linux. O LDP também suporta duas revistas on-line, Linux Gazette e Linux Focus. Toda a documentação se encontra à disposição gratuitamente via Internet. Discutimos os recursos LDP com um pouco mais de detalhes nas páginas 8 e 9.

29.2 Políticas e Procedimento

Ao pesquisarmos este capítulo, falamos com maiorais do mundo da administração de sistemas, da segurança em computadores, da comunidade de padrões e da legislação de informática. Ficamos surpresos que todos eles mencionaram que "políticas por escrito e assinadas" são essenciais para uma organização saudável.

As políticas e os procedimentos devem ser colocados no papel, aprovados pelas gerências e verificados por advogados. É preferível que essa preparação seja completada *antes* de os documentos precisarem ser usados para tratar de um problema espinhoso. Devem existir vários documentos de políticas:

- Políticas de serviços administrativos
- Direitos e responsabilidades dos usuários
- Políticas referentes a administradores de sistema (usuários com privilégios especiais)

3. http://www.zdnet.com/zdnn/stories/news/0,4586,2805929,00.html.

Snow666

- Política de contas de convidados

Procedimentos na forma de listas de verificação ou receitas podem ser usados para codificar a prática existente. Eles são úteis tanto para administradores de sistemas novos como para veteranos. Melhor ainda são os procedimentos na forma de scripts executáveis. Os diversos benefícios dos procedimentos-padrão são os seguintes:

- As tarefas são sempre feitas da mesma maneira.
- As listas de verificação reduzem a probabilidade de erros ou de etapas esquecidas.
- É mais rápido para o administrador de sistemas trabalhar com base em uma receita.
- As alterações são autodocumentáveis.
- Os procedimentos por escrito fornecem um padrão mensurável de correção.

Hoje, o Linux está substituindo os grandes mainframes do passado e realizando funções críticas no mundo corporativo. Em grandes instalações, as listas de verificação, muitas vezes chamadas de "livros executáveis", servem como documentação para tarefas comuns. Elas normalmente são mantidas on-line e também na forma de manuais impressos. Os administradores de sistemas que escrevem e mantêm os livros executáveis normalmente são uma camada à parte do pessoal de suporte que os usa, mas tal organização e padronização vale a pena a longo prazo.

Eis algumas tarefas comuns para as quais você pode querer configurar procedimentos:

- Adicionar um host
- Adicionar um usuário
- Localizar uma máquina
- Configurar envelopes TCP numa máquina
- Configurar backups para uma máquina nova
- Assegurar uma máquina nova
- Reiniciar um trecho de software complicado
- Reviver um site Web que não está respondendo ou não está servindo nenhum dado
- Descongestionar e reiniciar uma impressora
- Atualizar o sistema operacional
- Instalar um pacote de software
- Instalar software pela rede
- Atualizar software crítico (**sendmail**, **gcc**, **named** etc.)
- Fazer backups e restaurar arquivos
- Expirar fitas de backup
- Realizar desligamentos de emergência (todos os hosts, todos os hosts exceto hosts importantes etc.)

Muitas questões residem exatamente entre políticas e procedimentos. Por exemplo:

- Quem pode ter uma conta?
- O que acontece quando eles saem da empresa?

As resoluções de tais questões precisam ser colocadas por escrito de modo que você possa permanecer consistente e evitar tornar-se vítima do manjado e infantil subterfúgio "A mamãe disse não, vamos perguntar ao papai!". Muitas vezes, a porção "se" é a política e a porção "como" é o procedimento.

Algumas decisões políticas serão ditadas pelo software que você está usando ou pelas políticas de grupos externos, como os ISPs. Algumas políticas são obrigatórias caso a privacidade dos dados de seus usuários precise ser protegida. Chamamos esses tópicos de "política inegociável".

Acreditamos em particular que endereços Internet, nomes de host, UIDs, GIDs e nomes de usuários devem ser controlados por site. Alguns sites (empresas multinacionais, por exemplo) são claramente tão grandes para se implementar uma política dessas, porém se você conseguir fazê-lo, o gerenciamento válido para todo o site torna as coisas muito mais simples. Ferramentas que facilitam o gerenciamento de hosts e contas de usuários através de domínios administrativos estão disponíveis na rede. Nossas versões antigas, **addhost** e **adduser**, não são os exemplos mais adequados do gênero, porém eles ainda estão em uso e podem ser localizados em

ftp://ftp.cs.colorado.edu/sysadmin/utilities

Snow666

caso não encontre nada melhor.

Acreditamos piamente que logins *jamais* devam ser compartilhados. É muito mais fácil aplicar essa política caso a tentação de compartilhar logins seja eliminada. Costumávamos manter uma máquina convidada com uma política de criação de contas liberal como uma alternativa fácil ao compartilhamento clandestino, porém agora com contas de e-mail gratuitas disponíveis em várias fontes (AOL, Hotmail, Yahoo etc.) e terminais públicos em todo o canto (bibliotecas, cybercafés etc.), achamos que este serviço não é mais necessário.

Outras questões de política importantes que podem ter um escopo mais abrangente do que simplesmente seu grupo de administradores de sistema local são:

- Tratamento de brechas de segurança
- Controles de exportação de sistemas de arquivos
- Critérios para seleção de senhas
- Remoção de logins motivadas
- Material envolvendo direitos autorais (MP3s e DVDs, por exemplo)
- Pirataria de software

Manter bons canais de comunicação entre grupos administrativos num site grande pode evitar problemas e ajudar a desenvolver confiança e cooperação. Considere a possibilidade de realizar uma festa como um veículo de comunicação. Alguns grupos de administradores de sistema usam um MUD ou MOO tipo IRC como veículo de comunicação. Os papos podem rolar solto, mas se usados apropriadamente podem fazer com que as atividades da empresa caminhem de forma mais fluída, especialmente se existirem pessoas que trabalham fora da empresa ou em casa.

Políticas de segurança

O que você quer proteger? Seus dados? Seu hardware? Sua capacidade de recuperar rapidamente após uma quebra? Você deve considerar vários pontos de negociação ao projetar uma política de segurança de uma instalação:

- Serviços oferecidos *versus* segurança fornecida (mais serviços = menos segurança)
- Facilidade de uso e conveniência *versus* segurança (segurança = 1/conveniência)
- Custo de segurança *versus* risco (custo) de perda

A RFC2196, *Site Security Handbook*, é um documento de 75 páginas escrito em 1997 por um subgrupo do IETF (Internet Engineering Task Force). Ela alerta os administradores de sistemas sobre vários aspectos relacionados à segurança de sistemas, políticas de usuários e procedimentos. Ela não inclui uma receita para assegurar um site Internet, porém contém efetivamente algumas informações inestimáveis. As últimas 15 páginas são um conjunto maravilhoso de referências, tanto on-line quanto publicadas. A RFC2196 sugere que seus documentos sobre políticas incluam os seguintes pontos:

- *Determinar diretrizes* para hardware e software. Pode ser uma grande vitória envolver administradores de sistemas no processo de determinação destas diretrizes, pois eles muitas vezes conhecem defeitos de hardware, limitações de software e problemas de suporte que não são anunciados pelas equipes de marketing dos fabricantes.
- Uma *política de privacidade* que estabeleça expectativas referentes ao monitoramento dos e-mails e dos comandos dados pelos usuários e estabeleça políticas para lidar com arquivos de usuários.
- Uma *política de acesso*: quem pode ter acesso, o que eles podem fazer com seus acessos, que hardware e software eles podem instalar etc. Este documento deve incluir os mesmos alertas sobre uso desautorizado e monitoramento de linha que são inclusos na política de privacidade.
- Uma *política contábil* que diga claramente as responsabilidades tanto dos usuários quanto dos administradores de sistemas.
- Uma *política de autenticação* que estabeleça diretrizes para senhas e acesso remoto.
- Uma *política de disponibilidade* que descreve quando se espera que o sistema esteja ativo, lista horários de manutenção programados, dá instruções para relatar problemas e estabelece expectativas referentes a tempos de resposta.
- Uma *política de manutenção* que inclua regras sobre terceirização e especifique procedimentos para dar acesso ao pessoal de manutenção de terceiros.

www.snow666.blogspot.com
Snow666

Notadamente está faltando na RFC2196 uma política de autorização que especifique quem pode autorizar contas novas e privilégios estendidos. A RFC1244 original, *Site Security Handbook*, continha listas de questões concretas em vez de tipos de políticas, que poderia ser um pouco mais útil sob o ponto de vista do administrador de sistemas. A RFC mais nova inclui recomendações para cada tipo de serviço que uma máquina poderia rodar e descreve os problemas dos serviços e soluções em potencial.

Seja lá qual for a política adotada, ela deve ser explícita, por escrito, entendida e assinada por todos os usuários e administradores de sistemas. O cumprimento da política deve ser consistente, até mesmo quando os usuários são clientes que estão pagando por serviços computacionais. A falha na aplicação uniforme de políticas enfraquece sua validade legal e perceptível.

Acordos referentes a políticas para usuários

No departamento de Ciências da Computação da Universidade de Colorado, a política para usuários é emitida na forma de um shell inicial que imprime a política e exige que os usuários concordem com ela e a "assinem" antes de se tornar um verdadeiro shell para que possam usar suas contas. Esse esquema poupa tempo e dores de cabeça, porém consulte os *seus próprios* advogados antes de implementá-lo em sua instalação.

Eis algumas questões explícitas que devem ser tratadas num acordo sobre política para usuários:

- Compartilhar contas com amigos e parentes
- Executar programas para descobrir senhas[4] do sistema local
- Executar programas para descobrir senhas de outros sites
- Interromper serviços
- Invadir contas alheias
- Mau uso ou forjar e-mails
- Bisbilhotar arquivos de outros usuários (se puderem ser lidos? gravados? convidados?)
- Importar software da Internet (jamais? sempre? se o usuário verificar?)
- Usar recursos de sistema (impressoras, espaço em disco, modems, CPU)
- Copiar software licenciado
- Permitir que outros copiem software licenciado
- Copiar material que envolva direitos autorais (musica, filmes etc.)
- Conduzir atividades ilegais (fraude, difamação etc.)
- Envolver-se em atividades ilegais em alguns Estados, mas não em outros (por exemplo, pornografia)

Temos dois modelos de acordos para políticas de usuários em nosso site Web, www.admin.com. Um visa aos estudantes não graduados num laboratório onde um login é um privilégio e não um direito. Ele é o mais radical dos dois. O outro documento é para os professores, staff e estudantes graduados.

Como exemplo de um acordo sobre políticas simples e breve, incluímos o acordo que o departamento de Ciências da Computação da Universidade de Melbourne exige que os estudantes assinem para usarem os computadores da universidade:

Eu, abaixo-assinado, DECLARO PELO PRESENTE INSTRUMENTO que aceito e serei fiel às regras estabelecidas a seguir:

- *Usarei os recursos computacionais e de rede do Departamento exclusivamente para fins acadêmicos diretamente relacionados aos meus estudos das matérias de Ciências da Computação.*
- *Entendo que o Departamento concede contas de computador para uso exclusivo do concessionário. Portanto, não autorizarei ou facilitarei o emprego de minha conta ou arquivos por nenhuma outra pessoa, nem divulgarei minha senha para qualquer outra pessoa.*
- *Não acessarei ou tentarei ganhar acesso a nenhum computador, conta de computador, rede ou arquivos sem a devida e explícita autorização. Tal acesso é ilegal segundo as leis estaduais e federais e é contrário às normas da Universidade. Informarei imediatamente à Secretaria do Departamento de Ciências da Computação caso fique ciente de que tal acesso ocorreu.*

4. Por exemplo, **crack**, que é um programa para adivinhar senhas. Veja a seção 21.8 para obter maiores informações.

Snow666

- *Entendo que alguns softwares e dados residentes nos sistemas de arquivos que eu possa vir a acessar são protegidos por direitos autorais e outras leis e também por licenças e outros acordos contratuais; portanto, não violarei estas restrições.*
- *Não usarei os recursos da Universidade para obter, criar, executar ou distribuir cópias desautorizadas do software.*
- *Comprometer-me-ei a manter confidencial qualquer revelação a mim feita pela Universidade do software (inclusive métodos ou conceitos usados neste particular) licenciado à Universidade para uso em seus computadores e eu, por meio desse, salvaguardo e isento de quaisquer responsabilidades por perdas e danos a Universidade em relação a reivindicações de qualquer natureza que possam vir a surgir por motivo de qualquer revelação de minha parte a terceiros do dito software em violação deste comprometimento.*
- *Comprometo-me a manter o mais elevado padrão de honestidade e de integridade pessoal em relação ao emprego por minha parte dos recursos computacionais e de rede do Departamento. Garanto além disso que evitarei quaisquer ações em relação ao emprego por minha parte dos recursos computacionais e de rede do Departamento que possam vir a suscitar qualquer sintoma de má reputação ao Departamento ou à Universidade. Entendo que estou sujeito ao Regulamento 8.1.R7 da Universidade de Melbourne (descrito no Diário do Estudante), que também governa e regula emprego por minha parte dos recursos computacionais e de rede da Universidade. Entendo que o descumprimento de quaisquer dos princípios estabelecidos acima incorrerá em penas severas, inclusive reprovação numa matéria ou numa tarefa, suspensão ou revogação do direito de acesso aos recursos computacionais da Universidade, suspensão ou expulsão da Universidade, imposição de multas e/ou ação judicial, previstas no Crimes (Computer) Act 1988.*[5]

Atenção especial às palavras evasivas em relação à honestidade, integridade pessoal e a não suscitar qualquer sintoma de má reputação à Universidade. Exigências vagas como essas dão a você algum espaço de manobra posterior e ajudam a abranger quaisquer detalhes que possam ter sido deixados de fora da política. Embora o verdadeiro peso legal provavelmente seja insignificante, é uma boa idéia incluir tais exigências em seus acordos sobre políticas adotadas.

Acordos referentes a políticas para administradores de sistemas

Um documento sobre a política para administradores de sistemas (e outros com status especial) tem de estabelecer diretrizes para o uso de privilégios root e para honrar a privacidade dos usuários. É difícil responder à reclamação de um usuário de que o sistema de e-mail não está funcionando observando mensagens que voltaram. Porém uma cópia dos cabeçalhos normalmente é suficiente para caracterizar e resolver o problema.

Consulte a seção 3.4 para obter maiores informações sobre ***sudo****.*

Se o seu site usa uma ferramenta como o **sudo** para acesso root, é essencial que seus administradores de sistemas usem senhas fortes e não compartilhem seus logins com *ninguém*. Considere a possibilidade de executar regularmente **crack** sobre as senhas dos administradores de sistemas. Também é essencial que eles não executem **sudo tcsh** (uso simbólico de **sudo**), pois isso anula o recurso de registro de **sudo**.

Para alguns administradores de sistemas, a ânsia de demonstrar poderes root ultrapassa o bom senso. Sugira gentilmente carreiras alternativas a eles.

Em algumas instalações ter a senha root é símbolo de status, talvez mais valioso do que uma chave ao banheiro executivo. Normalmente, as pessoas que têm a senha são engenheiros que não precisam dela ou não deveriam possuí-la. Uma instalação que conhecemos ofereceu a senha root a todos os engenheiros, mas estipulou que todos que as possuíssem deveriam usar um bip e ajudar os outros quando necessário. As solicitações pela dita senha caíram vertiginosamente.

Uma outra técnica que usamos e que foi bem-sucedida é lacrar a senha root num envelope e escondê-lo num local conhecido pelo pessoal da administração de sistemas. Os administradores de sistemas geralmente usam **sudo** para realizarem seus trabalhos; caso efetivamente precisem da senha root por algum motivo, eles abrirão o envelope. A senha root é então alterada e um novo envelope, escondido. Não é difícil abrir um envelope usando vapor d'água quente, mas somente alguns administradores de sistemas possuem acesso físico ao local oculto, e confiamos no nosso staff quanto a respeitar o sistema.

5. Tenha em mente que essa é uma lei australiana, embora legislação similar referente a computadores e software tenha passado nos Estados Unidos. Veja a seção 29.3.

Snow666

Política e procedimentos para situações emergenciais

Decida com antecedência quem estará encarregado no caso de um incidente relacionada à segurança do sistema. Estabeleça uma cadeia de comando e mantenha fora de computadores ("off-line") os nomes e números de telefones dos cabeças. Pode ser que a melhor pessoa para ser encarregada disso seja um administrador de sistemas das trincheiras, não o diretor de TI (que normalmente é uma péssima opção para este papel). Guardamos um pequeno cartão laminado com nomes e números de telefones importantes impressos em tipo microscópico. Muito prático — e cabe na carteira de bolso.

Estamos acostumados a usar a rede para nos comunicarmos e acessarmos documentos. Entretanto, esses recursos poderão ficar indisponíveis ou comprometidos após algum incidente. Guarde todos os contatos e procedimentos relevantes off-line. Saiba onde encontrar as fitas de dump recentes e qual comando **restore** usar sem procurar em **/etc/dumpdates**. Evite falar com a mídia, especialmente se o incidente se der em tempo real.

Seqüestro de sites Web é a mais recente loucura no campo das quebras de segurança. Para um administrador de sistemas numa empresa que hospeda sites Web, um seqüestro pode ser um evento grave. Choverão ligações telefônicas dos clientes, da mídia, dos VIPs da empresa que acabaram de ver as notícias sobre o seqüestro na CNN. Quem atenderá a essas ligações? O que esta pessoa deverá dizer? Quem será o responsável? Que papel cada pessoa terá? Se estiver num ramo de alta visibilidade, realmente vale a pena pensar numa situação deste tipo, bolando algumas respostas pré-planejadas e, talvez, até mesmo realizar uma sessão prática para ajustar os detalhes.

Procedimentos para lidar com uma quebra de segurança são descritas no Capítulo 21, *Segurança*.

Planejamento em relação a desastres

Planejar para um desastre é melhor concretizado antes que o próprio desastre aconteça. Um fato infeliz em relação aos desastres é que a maioria deles ocorre nos laptops dos gerentes e, sob o ponto de vista do administrador de sistemas, eles têm o maior poder para alardear o fato. Nesta seção analisamos diversos tipos de desastres, os dados que você precisa recuperar graciosamente e os elementos importantes de um plano pós-desastre.

Há diversos tipos de desastres:

- Quebras de segurança: antes do ano 2000, cerca de 60% era originária de dentro da própria empresa. Em 2001, o aumento abrupto no número de ataques externos fez com que a porcentagem anterior caísse para algo em torno de 30%.
- Problemas de ambiente: surtos de tensão e falta de energia, falhas no sistema de refrigeração, inundações, furacões, terremotos, meteoros, ataques terroristas ou invasões de alienígenas.
- Erro humano: arquivos e bancos de dados eliminados ou danificados, perda de informações de configuração. (O seu sistema-espelho responde de forma tão rápida para impedir que um erro se propague para todo lado antes de você perceber o que aconteceu?)
- Panes de hardware espontâneas: servidores que não respondem, discos rígidos "fritos", redes defeituosas.

Em todas essas situações, você precisará acessar cópias on-line como cópias off-line de informações essenciais. As cópias on-line devem ser mantidas, se possível, numa máquina independente, uma que possua um conjunto relativamente rico de ferramentas, possua os ambientes-chave dos administradores de sistemas, rodem seu próprio servidor de nomes, possua um arquivo **/etc/hosts** local completo, não possua nenhuma dependência de compartilhamento de arquivos, possua uma impressora conectada etc. Eis uma lista prática da máquina de backup e na forma impressa:

- Uma descrição do procedimento pós-desastre: pessoas que devem ser acionadas, quando acioná-las, o que dizer.
- Números de telefone do contrato de serviços e números de clientes.
- Números de telefones locais importantes: staff, polícia, bombeiros, chefe, agencia de empregos.
- Dados sobre as configurações de hardware e software: versão do sistema operacional, níveis de correção, tabelas de partição, configurações de hardware de PC, IRQs, DMAs e coisas do gênero.
- Fitas de backup[6] e o cronograma de backups que as gerou.
- Mapas das redes.
- Números de série de software, dados de licenciamento e senhas.
- Informações para contato com o fornecedor para aquele disco de emergência que você precisa imediatamente.

6. Os backups podem ser citados; talvez você queira impor prazos de validade para suas imagens em backup.

Snow666

Uma suposição importante, mas muitas vezes não dita, feita na maioria dos planos pós-desastre é que o staff de administradores de sistemas estará disponível para tratar da situação. Infelizmente, as pessoas ficam doentes, têm festas de formatura, saem de férias, mudam de emprego e em momentos de estresse podem até mesmo se tornar hostis. Vale a pena considerar o que você deve fazer caso precise de ajuda emergencial extra. (Não ter administradores de sistemas suficientes à disposição pode, algumas vezes, por si só se constituir numa emergência caso seus sistemas sejam frágeis ou seus usuários não sejam muito dotados.)

Talvez você precise tentar formar uma espécie de pacto OTAN com uma empresa de consultoria local ou universidade que tenha talentos em administração de sistemas que possam ser compartilhados. Obviamente, você tem de estar disposto a retribuir colocando o seu pessoal à disposição caso eles venham a ter problema. Mais importante ainda, não opere próximo do limite em sua rotina diária. Contrate administradores de sistemas suficientes e não espere que eles trabalhem 12 horas por dia.

Teste o seu plano de recuperação pós-desastre antes de precisar usá-lo efetivamente. Se você acumulou um bom estoque para o bug do milênio, alguns itens (como lanternas) ainda devem ser úteis para desastres mais genéricos. Encontramos um tipo de lanterna muito interessante que se pode plugar na tomada. Enquanto a energia estiver ativa, ela permanece em plena carga. Quando a energia acaba, ela ilumina o ambiente de modo que você pode encontrá-la no escuro.

Teste seus geradores e no-breaks. Verifique se tudo com o que você se preocupa está ligado num no-break, que as baterias do no-break estejam boas e que o mecanismo de falha esteja funcionando. Para testar um no-break individual, basta desligá-lo da tomada. Para ter certeza de que o seu equipamento crítico esteja devidamente protegido pelo no-break, talvez você tenha de desativar os disjuntores e certificar-se de que sua configuração de emergência esteja realmente funcional como havia planejado.

A maioria dos problemas de alimentação é de curta duração, mas planeje para poder suportar por duas horas via bateria para ter tempo suficiente para desligar todas as máquinas apropriadamente no caso de uma falta de energia mais prolongada. Alguns no-breaks possuem uma porta serial ou interface Ethernet para você poder iniciar um desligamento adequado de máquinas não críticas após 5 minutos (tempo configurável) de falta de energia.

Tire proveito das interrupções de energia para fazer quaisquer atualizações rápidas que você havia planejado e que caibam dentro destes 5 minutos, como adicionar um disco a um servidor. O sistema está fora do ar de qualquer maneira, de forma que as pessoas já têm uma expectativa de serem incomodadas. Em algumas instalações, 5 minutos extras durante uma falta de energia é mais fácil do que aceitar um downtime programado com uma semana de antecedência. Se tiver máquinas antigas que você acredite que não estejam mais em uso, deixe-as desligadas até que alguém reclame. Pode ser que passe semanas — ou que jamais aconteça — dessa máquina "faltante" ser percebida.

Comsulte a seção 25.7 para obter maiores informações sobre problemas de ambiente.

Sistemas de refrigeração normalmente possuem um sistema de aviso que pode informá-lo caso a temperatura se torne muito elevada. Ajuste o valor de "muito alto" de forma que, após o sistema de refrigeração lhe enviar um bip, você tenha tempo suficiente antes de as máquinas realmente começarem a fritar; nós usamos 25° C em vez de 32° C, mas moramos nas montanhas a 45 minutos de distância (no verão, tempo indeterminado no inverno). Mantenha um termômetro mecânico ou operado por bateria na sala de máquinas — falta de energia significa perder todos aqueles graciosos indicadores eletrônicos que normalmente indicam a temperatura.

Uma instalação num grande órgão governamental americano recentemente construiu uma sala de máquinas extravagante e colocou um cluster Alpha de 256 nós nela para rodar enormes modelos científicos. Tudo estava ligado num no-break e todos os recursos eram o estado-da-arte. Infelizmente, uma simples falta de energia elétrica derrubou o centro por horas. Por quê? O PC que controlava o sistema de ventilação (ar-condicionado) não estava conectado ao no-break. Ele falhou e bagunçou todo o sistema de ar-condicionado. Teste cuidadosamente.

Informações variadas adicionais

Os provedores de acesso à Internet (ISPs) estão falindo, se juntando e sendo adquiridos numa velocidade incrível. Essas fusões demoliram os planos cuidadosamente elaborados de muitas empresas de manterem conexões redundantes com a Internet. Um ISP pós-fusão normalmente irá condensar circuitos que anteriormente pertenciam a empresas independentes. Clientes que antigamente tinham caminhos independentes para a Internet poderão passar a ter as duas conexões percorrendo um único conduíte e mais uma vez estarem à mercê de um único cabo de fibra ótica.

Quando a CNN ou a Slashdot anuncia que seu site Web está fora do ar, o mesmo efeito que faz com que o trânsito de uma rodovia fique lento provocado pela curiosidade dos motoristas em observar um acidente faz com que seu tráfego Internet aumente enormemente, muitas vezes a ponto de romper qualquer coisa que você havia corrigido. Se o seu site Web não é capaz de lidar com um aumento no tráfego na ordem de 25% ou mais, considere a possibilidade de fazer com que o seu software para balanceamento de carga direcione o excesso de conexões a um servidor

Snow666

que apresente uma única página dizendo: "Desculpe-nos, estamos com carga excessiva para poder atender à sua solicitação no momento".

*Consulte a seção 21.8 para obter maiores informações sobre **tripwire**.*

Use **tripwire** para se manter a par daquilo que seus administradores de sistemas estão fazendo, especialmente se grupos diferentes forem responsáveis por diferentes aspectos da mesma máquina. Correções em bancos de dados Oracle e correções em sistemas operacionais podem entrar em conflito entre si sem nenhum dos grupos perceber que ele deveria perguntar ao outro grupo o que está fazendo. A bisbilhotice via **tripwire** também é útil caso você trabalhe numa empresa que presta serviços de administração de sistemas e você se depare com a situação de ter de fazer uma limpeza após um período de administração de sistemas em que foi feito uma limpeza internamente pelo cliente e que não havia deixado muitos vestígios. Ele é capaz de identificar claramente o que foi alterado e quando, tornando mais fácil responder se a administração de sistemas local tentou jogar a culpa em você pelos erros dela própria.

29.3 Questões Legais

O governo federal e vários Estados norte-americanos possuem leis referentes a crimes relacionados à informática.

Em nível federal, há duas leis do início dos anos 90 e duas mais recentes:

- Federal Communications Privacy Act
- Computer Fraud and Abuse Act
- No Electronic Theft Act
- Digital Millennium Copyright Act

No início deste novo milênio, as grandes preocupações são a confiabilidade dos administradores de sistemas, operadores de redes e hospedagem de sites Web; criptografia reforçada para comércio eletrônico; questões referentes a direitos autorais e à privacidade.

Responsabilidades

Geralmente os administradores de sistemas não são responsabilizados pelo conteúdo armazenado pelos usuários nas máquinas pelas quais são responsáveis. Os ISPs têm, tipicamente, uma AUP (Appropriate Use Policy, política de uso apropriado) ditada pelos provedores acima e a exigem de seus clientes abaixo. Esse "fluxo de cima para baixo" de obrigações atribui responsabilidade pelas ações dos usuários aos próprios usuários, não ao ISP ou ao provedor acima do ISP. Essas políticas têm sido usadas para tentar controlar o spam (e-mail comercial não solicitado) e para proteger os ISPs em casos em que os clientes armazenam pornografia infantil em suas contas. Verifique as leis de sua região; estas podem variar muito.

Um bom exemplo (mas que é muito longo para ser incluído aqui) é a AUP usada pela mibh.net antes de ser adquirida pela Metromedia Fiber Network, Inc. A AUP incluía o jargão jurídico usual sobre ações ilegais, violações de propriedade intelectual e uso apropriado. Ela também incluía uma lista específica de atividades proibidas, bem como políticas de cumprimento, procedimentos para registrar reclamações e uma declaração referente a responsabilidades. Embora essa AUP não esteja mais sendo usada, ela é a melhor que vimos — uma cópia se encontra disponível em www.admin.com.

Criptografia

A necessidade de criptografia no comércio eletrônico e nas comunicações é clara. Entretanto, a criptografia é contra a lei em alguns países. Agências para cumprimento das leis não querem que cidadãos sejam capazes de armazenar dados que eles (a polícia) não conseguem decifrar.

Nos Estados Unidos, as leis referentes à criptografia estão mudando. No passado era ilegal exportar qualquer forma de tecnologia de criptografia forte. As empresas tinham de criar duas versões de software que incorporava criptografia: uma para vendas no mercado doméstico norte-americano e uma versão modificada para exportação. O absurdo patente dessa política (o resto do mundo possuía tecnologia de criptografia há um longo tempo) e as necessidades do comércio eletrônico motivaram o governo a alterar tal postura. Embora as restrições a exportações ainda não tenham sido totalmente eliminadas, a situação nos Estados Unidos está melhor do que costumava ser. Entretanto, os eventos terroristas de 11 de setembro de 2001 podem reverter o movimento do governo norte-americano em direção a menos restrições sobre a tecnologia de criptografia.

Um outro efeito colateral das leis norte-americanas anteriores é que vários projetos de desenvolvimento de software relacionado com criptografia estão locados em outros países. O IETF fez com que padrões funcionassem na área das comunicações seguras de ponta a ponta em nível de protocolo — a iniciativa IPSEC —, e os fabricantes

Snow666

começam agora a fornecer sistemas que a incluem. A parte referente à autenticação normalmente está embutida, porém a parte referente à criptografia em geral é instalada separadamente. Essa arquitetura preserva a flexibilidade para países nos quais a criptografia não pode ser usada.

Direitos autorais

As indústrias fonográfica e cinematográfica perceberam com alguma consternação que computadores pessoais são capazes de reproduzir música de CDs e de exibir filmes em DVD. Isto é, ao mesmo tempo, tanto uma oportunidade como uma ameaça para eles. O resultado final dependerá se essas indústrias responderão de uma forma pró-ativa ou reativa; infelizmente, parece que eles estão mais pendentes a adotar o caminho da reação.

O formato DVD usa uma chave criptográfica para bagunçar o conteúdo de um disco através de uma técnica chamada CSS (Content Scrambling System). A idéia era limitar a capacidade de reproduzir DVDs a aparelhos aprovados e licenciados. Os aparelhos de DVD dos consumidores incluem a chave de decodificação apropriada, assim como os reprodutores via software que vêm com a maioria das unidades de DVD para computadores.

Um estudante da Noruega aplicou engenharia reversa ao processo de criptografia CSS e postou um programa chamado DeCSS para a Web. O programa não evitava o esquema de criptografia dos DVDs; ele simplesmente usava a chave de descriptografia de um reprodutor Windows legítimo para decodificar o fluxo de dados do DVD e o salvava em disco para que pudesse executá-lo no Linux em vez de ter de usar o Windows.

A Motion Picture Association of America e A DVD Copy Control Association instauraram processos judiciais contra vários "distribuidores" do software DeCSS; todo o mundo que tivesse um site Web com link para uma cópia do DeCSS foi considerado um distribuidor. As ações alegavam que os acusados estavam envolvidos não em roubo de material com direitos autorais, mas sim na distribuição de segredos comerciais e no "logro de proteção de cópia" que se tornou ilegal nos Estados Unidos segundo o Digital Millennium Copyright Act of 1998. A MPAA ganhou a causa; DeCSS foi imputado como sendo uma violação do DMCA.

Dmitry Sklyarov, um cientista russo da computação, escreveu um programa para converter eBooks da Adobe para outros formatos, evitando as restrições que impediam a copia. Ele trabalhava para uma empresa na Rússia que produzia uma série de utilitários de conversão de formatos e de recuperação de senhas. O software, Advanced eBook Processor, teria permitido conversões como tornar um Adobe eBook acessível a usuários cegos na forma de um áudio book. Dmitry visitou os Estados Unidos e foi preso por violar um artigo do DMCA que proíbe a distribuição de tecnologia ou de informações que podem ser úteis em evitar restrições tecnológicas. Embora o caso ainda esteja pendente, ele foi libertado e teve permissão para retornar à Rússia.

Esses casos jurídicos oferecem uma variedade incrível de algumas das questões mais obscuras das leis de informática, de forma que eles estão sendo observados com grande interesse pela Electronic Frontier Foundation e pelos juristas desta área. Veja a excelente FAQ, Openlaw DVD/DeCSS, mantida por Rob Warren em www.cssfaq.org para mais opiniões e detalhes.

Durante a execução de vários casos, segundo o DMCA, foram encontrados muitos furos e arestas. Estes agora estão sendo "corrigidos" numa extensão do DMCA chamada SSSCA, Security Systems Standards and Certification Act, que agora está tomando rumo ao Congresso norte-americano. Essa lei vai tão longe que potencialmente bane a distribuição de qualquer código-fonte que seja coberto pela GPL — como o Linux.

Segundo a SSSCA, seria uma transgressão civil criar ou vender qualquer espécie de "dispositivo digital interativo" que não inclua e utilize tecnologias de segurança certificadas, aprovadas pelo governo federal. Um dispositivo digital interativo é definido como "qualquer hardware ou software capaz de armazenar, recuperar, processar, executar, transmitir, receber ou copiar informações na forma digital". Violações são delitos graves na esfera federal e incluem uma multa de até US$ 500.000. Esse projeto de lei dá à industria de informática um ano para criar "tecnologias de segurança certificadas" adequadas. Se ela não for bem-sucedida nesse intento, o Departamento de Comércio entrará em cena e estabelecerá o padrão. Nenhuma das reuniões dos representantes desse setor têm como exigência serem abertas ao público. Os proponentes da SSSCA também estão tentando amarrá-la a um acordo comercial que exigiria que outros países adotassem legislação similar para serem qualificados com status de comércio favorável.

A ACM (Association for Computing Machinery), a sociedade dos profissionais das Ciências da Computação, encaminhou uma carta ao senador Hollings, o proponente de tal projeto de lei, apontando que isso impediria as universidades de ensinar Computação Avançada, interferiria em dispositivos computacionais em tempo real que precisam de software simples e auditável (como aquele usado em controle de tráfego aéreo e monitoramento médico) e danificaria seriamente a segurança da infra-estrutura da Internet. Veja

http://www.acm.org/usacm/SSSCA-letter.html

www.snow666.blogspot.com
Snow666

para uma cópia completa da carta da ACM. A reação da EFF (Electronic Frontier Foundation's) foi mais forte ainda; veja www.eff.org.

Um outro exemplo de dificuldade de definição de padrões apropriados para propriedade intelectual e de direitos autorais na era digital é o caso da CyberPatrol. Essa empresa produz software para filtragem na Internet que grupos religiosos estão promovendo com pais, escolas e bibliotecas para protegerem as crianças de conteúdo censurável. Um canadense e um sueco criaram uma ferramenta chamada **cphack** que permitia que eles decifrassem a lista de bloqueio do software para ver exatamente quais sites Web estavam sendo bloqueados, o nível de taxa de erro e que listas não tão óbvias poderiam estar presentes. Eles relataram, por exemplo, que qualquer um que tivesse criticado o software estava bloqueado em todas as categorias.

Mattel, que na época era dono da CyberPatrol, processou os criadores da ferramenta, reivindicando que a licença da CyberPatrol proibia engenharia reversa. Mattel obteve um mandado preliminar contra a distribuição do software, mas infelizmente o caso jamais foi a julgamento; ele foi solucionado pouco antes de ir a julgamento. Os autores da ferramenta a venderam por US$ 1 e fizeram um acordo judicial aprovado pelo poder público. Parece que os autores se renderam (honorários de advogados à parte!), porém analisando de mais perto, parece que Mattel estava tentando garantir a propriedade de uma ferramenta que foi originalmente lançada segundo a Licença Pública GNU.

Mattel esperava usar seus recém-adquiridos direitos sobre propriedade intelectual para impedir que a ferramenta **cphack** fosse copiada via Internet (como se isso alguma vez funcionasse). Pelo fato dos autores de **cphack** a terem lançado segundo a GPL, a distribuição irrestrita do programa original é permitida mesmo que Mattel possua o direito autoral sobre ele. Assim que um software for publicamente lançado sob termos liberais, como os contidos na GPL, não dá para voltar atrás.

Privacidade

A privacidade sempre foi difícil de ser salvaguardada, porém com o advento da Internet, ela está em perigo mais do que nunca. Por exemplo, durante um recente incidente na Universidade de Michigan, os boletins médicos dos pacientes no sistema de atendimento médico da Universidade de Michigan foram inadvertidamente publicados na Internet. Os dados ficaram disponíveis à vontade por meses até um estudante ter percebido o descuido.

Um outro grande escândalo referente à privacidade, este intencional, envolveu a DoubleClick.net, uma agência de propaganda que cria a maioria dos anúncios via banners nas páginas Web. A DoubleClick prometeu por anos que os usuários em seu sistema jamais seriam rastreados ou identificados. Recentemente, porém, eles compraram uma empresa que faz garimpagem de dados e começa a coletar dados de cada usuário que visitou uma página Web contendo um anúncio da DoubleClick. O furor resultante fez com que a DoubleClick revogasse o projeto por enquanto e contratasse dois advogados de peso para cargos na área de "privacidade" para descobrirem uma maneira da empresa espreitar legalmente os usuários que recebem seus anúncios.

A DoubleClick é café pequeno quando comparada a uma nova ameaça à privacidade por meio da combinação de nossos ISPs e uma empresa chamada Predictive Networks. De acordo com o PRIVACY Forum Digest, a Predictive, com a ajuda dos ISPs, planeja coletar os URLs que cada um de nós visita, as palavras-chave que digitamos nos mecanismos de busca e outras informações observando nossa navegação na Web. Com base nesses dados, eles construirão uma "assinatura" digital de cada um de nós para usá-la como perfil para endereçar conteúdo Internet e anúncios específicos para cada usuário.

A Predictive diz que nossas informações são "anônimas" e que podemos confiar em cada um dos envolvidos: os funcionários da Predictive, os funcionários dos ISPs, os anunciantes, os provedores de conteúdos — todo o mundo. Podemos solicitar uma cópia de nossa assinatura digital, mas teríamos de pagar por ela. Podemos também optar por ficar fora desse "serviço", mas nossa conectividade com a Internet através desse ISP poderia custar mais ou ser rescindida. Na época em que este livro era escrito, o site Web da Predictive (www.predictivenetworks.com) ainda não possuía nenhuma política de privacidade e não muitas informações sobre o que eles realmente fazem. O artigo PRIVACY Forum Digest (V09, #13, www.vortex.com) inclui mais detalhes sobre os planos e a posição da Predictive em relação à privacidade dos usuários da Internet.

Impingindo uma política

Os arquivos de log podem provar a você que está em dúvida de que a pessoa X fez a coisa errada Y, mas para um tribunal tudo isto é apenas evidência rumorosa. Proteja-se com políticas por escrito. Os arquivos de log algumas vezes exibem horários, que são úteis, mas não necessariamente evidências admissíveis a menos que o seu computador esteja rodando o NTP (Network Time Protocol) para manter o relógio dele sincronizado com a realidade.

Talvez você precise de uma política de segurança para processar alguém por uso indevido. Ela deve incluir uma declaração do tipo: *O uso desautorizado dos sistemas computacionais da Universidade pode envolver não so-*

Snow666

mente a transgressão da política da Universidade, como também uma violação das leis estaduais e federais. O uso desautorizado é um crime e pode envolver penalidades criminais e civis; ele será processado com todo o rigor da lei.

Aconselhamos que você coloque um alerta em **/etc/motd** (a mensagem do arquivo diário) que informe os usuários sobre sua política de "xereta". O nosso diz o seguinte:

```
Suas entradas feitas via teclado poderão ser monitoradas no caso de um incidente de segurança
perceptível ou real.
```

Algumas conexões (como as sessões **ftp**) não visualizam a mensagem do dia. Os usuários também podem suprimir a mensagem criando um arquivo chamado **.hushlogin** em seus diretórios de origem. Talvez você queira garantir que os usuários vejam a notificação pelo menos uma vez incluindo-a nos arquivos de inicialização que você dá a usuários novos. Caso precise usar **ssh** para fazer o login (e você realmente deve usá-lo), configure então **sshd.config** para sempre exibir o arquivo **motd**.

Certifique-se de especificar que os usuários pelo simples ato de usarem suas contas aceitam a política por escrito imposta por você. Explique onde os usuários podem obter cópias adicionais dos documentos da política e envie documentos-chave num boletim apropriado. Inclua também a penalidade específica pelo descumprimento da política (eliminação da conta etc.).

Suponha que algo impróprio seja postado para um grupo de notícias ou para a Web pelo seu site. Se você for a CompuServe (agora parte da AOL), isso é um problema. Num caso chamado *Cubby versus CompuServe*, algo difamatório foi postado. O juiz determinou que a CompuServe não era culpada, porém definiu que o moderador do grupo de notícias para o qual ele foi postado foi negligente. Quanto mais você tenta controlar informações, mais responsável se torna.

Esse princípio é maravilhosamente ilustrado pela história de uma empresa texana fundada por um de nossos estudantes empreendedores, Cheeser. Ele escreveu scripts Perl para garimpar os grupos de notícias Usenet, coletar imagens perniciosas e construiu um site Web pago baseado neste conteúdo. Ele cobrava US$ 12/mês dos assinantes e estava ganhando muito dinheiro com isso.

Cheeser tentou ser um pornógrafo responsável e não se inscreveu em grupos de notícias sabidamente de pedofilia. Ele também monitorava vários grupos de notícias que estavam na crista da onda, algumas vezes com conteúdo ilegal, outras não. Esse pequeno descuido e sua opção de sediar seu negócio num Estado tão conservador quanto o Texas o levaram à ruína.

Baseada numa denúncia anônima (talvez de um concorrente), a polícia local confiscou os seus computadores. Certamente eles encontraram algumas amostras de pornografia infantil que havia sido postada a um dos grupos de notícias "seguros'. O caso criminal jamais foi a julgamento, porém durante a negociação da contenda ficou claro que o juiz achava Cheeser culpado — não somente por ter criado aquele tipo de conteúdo, como também por não ser um censor suficientemente bom. A implicação era de que se Cheeser não tivesse feito nenhuma espécie de censura, ele estaria legalmente certo. Jamais censure sua pornografia.

Esse princípio também se aplica às demais interações com o mundo externo. Sob um ponto de vista legal, quanto mais você monitora o uso da Internet por parte de seus usuários, maior o seu nível de envolvimento e de responsabilidade pelas ações ou postagens de seus usuários. Caso saiba de alguma atividade ilegal ou acionável, você tem uma incumbência legal de investigá-la e de relatar suas descobertas à justiça.

Por essa razão, alguns sites limitam os dados que eles logam, o período de tempo no qual estes arquivos de log são mantidos e a quantidade de histórico de log mantido em fitas de backup. Alguns pacotes de software (por exemplo, o cache de Web Squid) ajudam na implementação dessa política incluindo níveis de registro que auxiliam o administrador de sistemas a depurar problemas, mas sem violar a privacidade do usuário.

Os administradores de sistemas devem estar familiarizados com todas as políticas relevantes da empresa ou da universidade e devem garantir que tais políticas sejam cumpridas. Políticas inconsistentes e não impostas são piores do que nenhuma política em si, tanto sob o ponto de vista prático quanto legal.

Licenças de software

Muitos sites pagaram por K cópias de um pacote de software e têm N cópias em uso diariamente, onde K < N. Ser pego em tal situação poderia ser prejudicial à empresa, provavelmente mais prejudicial do que o custo dessas N-menos-K licenças adicionais. Outros sites recebem uma cópia de demonstração de um pacote de software caro e o burlam (reinicializam a data na máquina, descobrem o código de licenciamento etc.) para fazer com que ele continue a funcionar após a expiração do período de demonstração. Como um administrador de sistemas deve lidar com casos de violação dos acordos de licenciamento e fazer cópias do software em máquinas que não possuem licença? O que

Snow666

fazer ao descobrir que máquinas pelas quais você é responsável estão rodando software pirata? O que dizer de shareware que jamais foi pago?

É uma questão muito difícil. Seus superiores normalmente não o suportarão em suas solicitações de que cópias não licenciadas de software sejam eliminadas ou pagas. Normalmente, é um administrador de sistemas que assina o acordo para eliminar as cópias de demonstração após uma determinada data, porém é um gerente que toma a decisão de eliminá-las.

Até mesmo se o trabalho for o melhor que você já fez, suas integridades pessoal e profissional estão na mira. Felizmente, no mercado de trabalho atual, a demanda por administradores de sistemas bem qualificados está em alta e a sua recolocação será rápida. Sabemos de vários casos nos quais o gerente imediato de um administrador de sistemas não queria se envolver na situação e disse ao administrador de sistemas para não complicar mais a situação. O administrador escreveu então um memorando ao chefe pedindo para corrigir a situação e documentando o número de cópias do software que estavam licenciadas e o número de cópias em uso. O administrador pegou algumas frases da licença de uso e mandou cópias do memorando para o presidente da empresa e para os gerentes de seu chefe. Em um caso, esse procedimento funcionou, e o gerente do administrador de sistemas foi mandado embora. Num outro caso, o administrador de sistemas pediu demissão quando até mesmo a gerência mais alta se recusou a fazer a coisa certa.

Software não licenciado no mundo Linux de fonte aberto não é tanto problema[7]. Os autores de software muito interessantes normalmente os dão. Algumas vezes eles solicitam uma pequena contribuição ou um belo CD de música caso você use seus pacotes extensivamente.

Spam: e-mail comercial não solicitado

Anunciantes, pessoal de marketing e artistas coadjuvantes pipocaram aos montes na Internet para tirarem vantagem da comunicação "gratuita" oferecida pelo e-mail. O custo de enviar um anúncio num e-mail-propaganda para milhares de pessoas é minúsculo quando comparado ao custo de envio de mala-direta tradicional (em papel) e a taxa de resposta aparentemente é melhor. Essa situação cria dois grandes perdedores: os consumidores, que têm de ficar se livrando de montanhas de spam todos os dias e os ISPs, que pagam pelo tráfego em suas redes.

Não entraremos nos detalhes técnicos do rastreio do spam e na luta contra spams aqui. Esse assunto é coberto de forma mais aprofundada no Capítulo 19, *E-Mail*. Entretanto, mencionaremos alguns dos aspectos legais.

Os Estados Unidos possuem leis, a maior parte delas em nível estadual, que foram usadas com sucesso para processar os remetentes de spam. Pelo menos em um caso, os remetentes foram obrigados a pagar por cada mensagem de e-mail enviada, pois o spam interferia nas operações normais de uma empresa. Infelizmente, a maioria dos receptores de spam simplesmente os elimina e não se importa em tentar rastrear o spammer (remetente) e retaliá-lo.

Os ISPs realmente tentam evitar que os spammers usem seus recursos, não somente devido à largura de banda que utilizam, como também pelo fato de o spam violar tipicamente a política de uso apropriado do provedor que está acima deles e, portanto, de os colocarem em risco de perderem a própria conexão de rede.

Você pode encontrar uma prática página contendo links para recursos e legislação referentes a spam em

http://www.elsop.com/wrc/nospam.htm

John Gilmore, um dos fundadores da Electronic Frontier Foundation, detém uma visão alternativa sobre spam. Ele acha que a defesa da primeira emenda é muito mais importante do que ficar lutando contra spam e que qualquer um deve ser capaz de enviar spam sempre que quiser. Ele possui scripts que detectam um único emitente enviando muito e-mail e os desativa, agindo como um receptor e não tentando legislar o que pode ou não ser enviado. Os scripts reduzem a velocidade dos emitentes depois de 50 mensagens terem sido transmitidas através de sua máquina transmissão aberta e os interrompe após 100 mensagens.

John acredita que tentar tornar o spam algo ilegal é tecnicamente infactível e incorreto; o spam tem de ser filtrado na ponta receptora. Ele também acha que AUPs (Appropriate Use Policies, políticas de uso apropriado) são uma restrição ilegal de comércio. Ele propõe um interessante cálculo do valor de várias mídias em relação ao número de usuários:

- TV: o valor da rede é n, o número de telespectadores
- Telefone: o valor é n^2, o numero de pares de participantes possíveis
- E-mail: o valor da rede é n!, o número de grupos possível

7. Software não licenciado é um tanto raro. Software licenciado sob a Licença Pública GNU é outra coisa! Essa licença garante liberdades que não podem ser confundidas com falta de licença. Ela inclusive obriga a garantia da liberdade a todos que receberem tal software ou derivados dele (N.R.T.).

Snow666

Pesquisas de administradores de sistemas

A SAGE (System Administrators Guild, Comunidade dos Administradores de Sistema), associada à USENIX Association e o SANS (System Administration, Networking and Security Institute, Instituto de Administração, Segurança e Interconexão de Sistemas) realizam pesquisas anuais que se concentram principalmente nos salários dos administradores de sistemas. Os resultados dessas pesquisas para o ano 2000 (o último disponível na época em que este livro era impresso) são apresentados de forma resumida nas próximas seções. As pesquisas não fazem distinção entre administradores de sistemas Linux e UNIX, mas diferencia UNIX do Windows.

Pesquisa salarial feita pela SAGE

A SAGE realizou uma pesquisa salarial para administradores de sistemas em sua conferência LISA e via Web. O relatório completo se encontra disponível em

http://sageweb.sage.org/jobs/salary_survey/

Os membros da SAGE podem fazer o download de um arquivo no formato Acrobat fornecendo seus números de sócio e senha; os demais têm de se inscrever e, então, recebem o arquivo via e-mail[8]. A SAGE na verdade não vende o seu endereço, mas talvez lhe envie informações sobre suas conferências e publicações. (Na verdade, você pode optar por não receber o spam deles ao se inscrever.)

A seguir alguns dados interessantes da pesquisa de 2000, que incorporam respostas de mais de 5.000 administradores de sistemas que participaram da conferência ou preencheram o formulário de pesquisa na Web. A maioria dos que responderam trabalhava em período integral como administrador de sistemas e considerava esta função como sua fonte primária de renda. Cerca de 80% era dos Estados Unidos; o restante de outros 68 países diferentes.

- O salário médio foi cerca de US$ 70.000 nos Estados Unidos, a menos que se trabalhasse numa universidade, onde o salário era cerca de US$ 12.000 a menos.
- Mais de 60% dos que responderam receberam aumentos em 2000 (abaixo dos 86% em 1999), variando entre 11% para os que permaneceram no mesmo cargo a 17% para aqueles que mudaram de cargo e de empresa.
- Mais de 70% das organizações não consegue encontrar administradores de sistemas suficiente — a caça continua.
- Cerca de metade recebeu bônus; 15% recebem pagamento de horas extras.
- Os administradores de sistemas assalariados trabalham, em média, 46,7 horas por semana.
- Os sistemas operacionais mais comuns foram (em ordem): Solaris (70%), Windows NT (67%), Linux (62%) e Windows 95 e 98 (49%). A maioria dos administradores de sistemas lida com ambientes heterogêneos contendo uma média de 5 sistemas operacionais diferentes.
- A formação acadêmica não se correlaciona exatamente com o salário recebido. Um diploma em Ciências da Computação vale somente US$ 6.000 em relação a um diploma de segundo grau.
- Anos de experiência efetivamente se correlacionam com o salário; a maioria dos administradores de sistemas tinha de 5 a 10 anos de experiência.
- Mais de 75% dos que responderam à pesquisa esperavam continuar como administrador de sistemas pelo menos por mais 5 anos. Essa é uma mudança boa; o cargo de administrador de sistemas costumava ser visto como uma função que você fazia enquanto aguardava ser promovido para desenvolvedor de software. Porém alguns profissionais *gostam* de ter mais mudanças de contexto por dia do que um desenvolvedor tem num ano de trabalho.
- Menos do que 9% dos administradores de sistemas eram mulheres, 13% a menos do que em 1999. Os salários das mulheres era cerca de US$ 5.000 mais baixos do que os dos homens, mas no Reino Unido os salários eram cerca de US$ 5.000 mais altos. Legal!
- Quase 50% dos administradores de sistemas estavam na faixa etária de 25 a 35 anos e menos do que 1% abaixo de 20 ou acima de 55.
- As partes mais chatas e problemáticas da vida de um administrador de sistemas eram (em ordem): lidar com o gerenciamento, carga horária exagerada e políticas e burocracia dentro da empresa.

8. O acesso às pesquisas parece ter sido liberado, em novembro de 2003. Foi possível ver o arquivo (PDF) sem a necessidade de inscrição. A última pesquisa disponível é referente ao ano de 2002, juntamente com uma compilação dos principais comentários feitos pelos administradores de sistema. Leitura recomendada (N.R.T.).

Snow666

Ao analisar o formulário de pesquisa, a maioria dos administradores de sistemas percebeu que eles realmente não sabiam exatamente como despendiam o dia. A maioria das instalações sente que tem um número de staff menor do que seria necessário. Não é preciso mais do que uma meia dúzia de usuários chorões para tornar a relação usuário/administrador de sistemas opressiva.

Pesquisa salarial elaborada pelo SANS

A pesquisa do SANS para o ano 2000 foi administrada via Web e incluiu cerca de 7.000 administradores que responderam. O associado do SANS normalmente é menos técnico que o pessoal da SAGE e inclui mais administradores de sistemas que lidam basicamente com desktops Windows. Os que responderam foram classificados como administradores de sistemas, administradores de rede, administradores de segurança, administradores de bancos de dados, consultores em segurança de sistemas ou auditores de segurança de sistemas.

Muitas das perguntas feitas nesta pesquisa coincidiram com as da pesquisa feita pela SAGE, mas comparações diretas são difíceis devido a uma gama mais ampla de participantes. Os salários foram subdivididos por região; incluímos os dados referentes aos Estados Unidos no resumo a seguir.

Daqueles que responderam à pesquisa, cerca de 50% se rotulavam como administrador de sistemas e outros 24% como administradores de rede. As demais categorias eram porcentualmente bem menores. Eis alguns dos resultados mais interessantes:

- A representatividade dos sistemas operacionais era praticamente a metade para UNIX/Linux e outra metade para o Windows. Do pessoal do UNIX/Linux, 60% era Solaris.
- A média salarial era cerca de US$ 65.000 no geral: US$ 70.000 para o pessoal UNIX/Linux e US$ 60.000 para os administradores de sistemas Windows. As cifras para as mulheres eram mais baixas tanto em termos de salário quanto de aumentos. Os administradores de segurança se deram melhor do que os de sistema e os administradores de rede tiveram um desempenho pior. Os aumentos foram de 11% em média.
- Um diploma universitário em Ciências da Computação valia cerca de US$ 5.000 a mais para um administrador de sistemas NT e de US$ 8.000 para UNIX; um grau de mestre acrescentava cerca de US$ 5.000 a US$ 8.000. Os que possuíam curso de mestrado ganhavam mais do que os que tinham Ph.D.

29.4 Escopo do Serviço

Os serviços fornecidos por um grupo de suporte administrativo têm de ser claramente definidos, caso contráario as expectativas dos usuários não condirão com a realidade. Eis algumas questões a serem consideradas:

- Tempo de resposta
- Serviço nos fins de semana e fora do horário
- Telefonemas de casa (suporte para máquinas pessoais dos usuários)
- Hardware estranho (ou do tipo)
- Hardware antigo
- Sistemas operacionais suportados
- Configurações-padrão
- Expiração das fitas de backup
- Software de propósito especial
- Tarefas de zelador (limpeza de telas e de teclados)

Além de saber quais serviços são fornecidos, os usuários também têm dê conhecer o esquema de prioridade usado para gerenciar a fila de tarefas. Os esquemas de prioridade sempre têm espaço variável, mas tente elaborar um que cubra a maioria das situações com poucas ou nenhuma exceção. Algumas variáveis relacionadas com prioridade são:

- O número de usuários afetados
- A importância dos usuários afetados
- O barulho feito pelos usuários afetados (rodinhas das cadeiras guinchando)
- Importância dos prazos (tarefa de casa até tarde da noite *versus* proposta de concessão de pesquisa que financia parcialmente o grupo de administração de sistemas)

Nosso grupo de suporte para a faculdade, staff e alunos de computação elaboraram um conjunto de documentos que delineia seus serviços, esquema de prioridade e mecanismos de contato. Esse grupo de clientes contém vá-

Snow666

rios níveis de importância e de repercussão. Os documentos referentes às políticas têm sido usados há alguns anos com bons, mas não perfeitos, resultados. Há cópias disponíveis em www.admin.com.

Os usuários geralmente partem do pressuposto que seus dados importantes são armazenados em fitas de backup que serão arquivadas para sempre. Mas as fitas de backup não são eternas; a mídia magnética tem tempo de vida finito após o qual a leitura de dados se torna difícil. As fitas de backup também podem ser intimadas; pode ser que sua empresa não *queira* que dados antigos fiquem disponíveis para sempre. É melhor manter uma política explícita por escrito que especifique por quanto tempo as fitas de backup serão mantidas. Tal política ajuda os usuários a terem expectativas realísticas e o protege de ter de retornar muito tempo no passado para atender advogados.

Numa instalação Linux, cada usuário tem acesso ao código-fonte e pode facilmente criar um kernel personalizado para sua própria máquina. Você tem de considerar essa possibilidade em sua política corporativa e provavelmente deve tentar manter uma padronização com algumas poucas configurações de kernel específicas. Caso contrário, seu objetivo de manutenção e de expansibilidade com a empresa encontrará sérios obstáculos. Encoraje seus funcionários criativos e que gostam de fuçar o sistema operacional para que sugiram modificações de kernel de que eles precisem para suas funções. Faça com que eles tomem parte do processo de padronização para que você não tenha de fazer a manutenção de kernels distintos para cada engenheiro.

29.5 Sistemas de Relatos de Problemas

Nosso sistema de relatórios de problemas usa um e-mail alternativo chamado "trouble". Certa vez fomos bombardeados com relatos de problemas que eram incompletos ou incompreensíveis. Criamos um script que fazia ao usuário perguntas específicas como:

- Em que host ocorreu o problema?
- O problema pode ser reproduzido?
- Qual o nível de importância deste problema ser corrigido imediatamente?

A rebelião de usuários se iniciou uma hora depois e dentro de um dia retornamos o sistema ao que era antes. Seu único valor parece ter sido aquele relacionado com o furor em relação ao script; na verdade, muitos usuários leram as perguntas que estávamos fazendo, e a qualidade de nossos relatos de problemas de formato livre aumentou.

Uma outra instalação tratou deste problema enviando uma mensagem que explicava quais informações eram importantes num relato de problema e deu exemplos de relatos inúteis. Quando um relato inútil era recebido, ele era respondido com uma desculpa ("Desculpe-nos, não temos informações suficientes para ...") junto com uma cópia do próprio relato. Os usuários se ligaram rapidinho.

É preciso usar algum tipo de sistema de bilhetagem de problemas, caso contrário os problemas receberão cinco respostas (as fáceis) ou nenhuma (as difíceis). O sistema deve registrar em log os relatos de problemas solucionados e talvez enviar uma cópia da solução para administradores novatos e trainees. Os arquivos de log se tornam uma fonte inestimável de dados para ser explorada durante a preparação de relatórios para os gerentes (especialmente quando você precisa justificar a contratação de mais pessoal para a administração do sistema) e as cópias para os administradores novos permite que eles se familiarizem com os problemas mais comuns de sua instalação e as respectivas soluções. Eles também demonstram o tom apropriado a ser usado nas mensagens aos usuários.

Nosso sistema de bilhetagem é o **wreq** (www.math.duke.edu/~yu/wreq). O **wreq** baseia-se no **req** da Universidade de Maryland, ao qual ele acrescenta funcionalidade extra e uma interface para a Web. Ele tem praticamente tantos recursos quanto o produto comercial disponível Remedy, porém é mais fácil de ser configurado e usado e mais fácil de caber no seu orçamento (gratuito!).

Os gráficos de tíquetes submetidos, solucionados e em estado de putrefação do **wreq** são especialmente úteis tanto para os administradores de sistemas quanto para os gerentes. Eles podem ajudá-lo a detectar tendências de longo prazo. As principais desvantagens do **wreq** são sua documentação pobre (quase beirando a inexistência) e a sua falta de escalabilidade.

29.6 Gerenciando a Gerência

É essencial que seus gerentes o respeitem e o apóie. O suporte da gerência em relação a políticas de segurança rígidas algumas vezes é mais difícil de obter. Segurança rígida invariavelmente significa inconveniência para os usuários, e os usuários normalmente o superam em número e poder de reclamação. Certifique-se de que qualquer mudança na política de segurança que atinja usuários (mudar de **telnet** para **ssh**, conversão de senhas para chaves DSA etc.) seja anunciada com antecedência, seja bem documentada e também seja bem suportada no momento da mudança. A documentação deve ser fácil de ser entendida e deve fornecer receitas de bolo para lidar com o novo sistema. Re-

Snow666

serve horas extras para o staff quando mudar para o novo sistema de modo que possa lidar com os usuários assustados por não terem lido seus e-mails ou o **motd**.

A alta gerencia normalmente não tem a mínima idéia do que os administradores de sistemas fazem. Manter um diário por uma semana que registre o que você faz e quanto tempo isto leva irá surpreender até a você mesmo. Esse tipo de documentação é essencial em suas campanhas reivindicando pessoal ou equipamento extra. Ele também pode ser uma fonte de poder nas batalhas políticas do dia-a-dia. É sensato manter bons registros mesmo na ausência de um objetivo em particular.

Os gerentes, especialmente os gerentes não técnicos, normalmente estão bem longe da realidade em suas estimativas da dificuldade de uma tarefa ou da quantidade de tempo que levará para completá-la. Isso é particularmente verdadeiro nas tarefas de diagnóstico.

Tente estabelecer expectativas reais. Duplique ou triplique suas estimativas de tempo para tarefas grandes ou cruciais. Se uma atualização for feita em dois dias em vez de três, a maioria dos usuários irá lhe agradecer em vez de xingá-lo como eles normalmente o fariam caso sua estimativa tivesse sido de um dia.

Algumas vezes é difícil para um administrador de sistemas colocar em prática uma política por escrito. Por exemplo, "Temos 8 licenças para o Excel e 47 cópias instaladas". Solicite verba para adquirir mais cópias. Se isso falhar, escreva um memorando documentando o problema com cópia para a alta gerência. Felizmente, a demanda por bons administradores de sistemas está em alta, de modo que sua procura por um novo emprego deve ser curta.

29.7 Contratação, Demissão e Treinamento

Há duas formas de montar uma equipe de administradores de sistemas:

- Contratar gente experiente.
- Cultivar a sua própria equipe.

O pessoal experiente normalmente engrena mais rapidamente, mas você sempre terá de fazer eles desaprenderem certas coisas. Para fazerem seus trabalhos, precisarão de acesso root. Mas você ainda não os conhece e não vai querer se arriscar colocando imediatamente os dados da empresa em suas mãos.

Envolve um bom tempo e empenho para treinar um administrador de sistemas, e as redes de produção não são um terreno ideal para treinamento. Mas dada a pessoa certa (inteligente, interessada, curiosa, cuidadosa etc.), o resultado final normalmente é melhor.

Desenvolvemos duas ferramentas de avaliação para pretendentes ao cargo experientes. Costumávamos chamá-los de "testes", mas descobrimos que algumas instituições não podem aplicar testes em candidatos a um emprego. Não testamos mais; estimamos e avaliamos.

A primeira avaliação escrita (não teste) solicita aos candidatos que expressem sua experiência e conhecimento em várias tarefas de rede e de sistemas. A escala de familiaridade varia de 0 a 4:

- Jamais ouvi falar a respeito (0)
- Ouvi falar a respeito mas jamais fiz (1)
- Já o fiz, poderia fazê-lo sob supervisão (2)
- Poderia fazê-lo sem supervisão (3)
- Conheço-a bem, poderia ensinar alguém a fazê-lo (4)

No meio das perguntas há diversos alarmes. Por exemplo, na seção de hardware há uma pergunta sobre conectores RS-232 seguida por uma sobre "conectores MX".[9] Essas pegadinhas permitem avaliar o fator Besteira nas respostas do candidato. Um 3 nos conectores MX seria suspeito. Após o questionário, você poderia perguntar inocentemente, "Bem, para que *você* usa conectores MX?".

A segunda avaliação é desenhada para uso durante uma entrevista telefônica. As perguntas são feitas para extrair respostas rápidas de candidatos que realmente conhecem o assunto. Damos +1 para uma resposta correta, 0 para um "Não sei" e -1 para Besteira óbvia ou para quem digitar **man xxx** ao fundo.

Esses dois esquemas têm sido uma boa métrica para nós. A porcentagem de pegadinhas que usamos é determinada pelo nosso pessoal de contratação; uma ou duas perguntas destas não bastam. Tenha em mente que estas avaliações não tratam de algumas das mais importantes questões referentes aos administradores de sistemas a serem contratados:

9. MX se refere a um registro para um servidor de e-mails no DNS, não é uma conexão serial.

Snow666

- Eles se darão bem com os demais membros da equipe?
- Como é o relacionamento deles com os usuários?
- Eles aceitarão orientação?
- Eles se encontram numa curva de crescimento com uma inclinação positiva?

Essas avaliações que usamos podem ser encontradas on-line em www.admin.com. Você é capaz de localizar todas as pegadinhas?

Uma entrevista pessoal pode responder a algumas dessas questões. Uma conversa telefônica com referências normalmente lhe dirá muito mais. Ouça atentamente, muitas pessoas não gostam de dizer coisas ruins a respeito de um antigo empregador ou companheiro de trabalho, de forma que eles encontram uma série de maneiras de lhe dizer (caso você esteja ouvindo atentamente) que um funcionário em potencial tem problemas. Suspeite se o candidato não incluir empregadores recentes como referências.

Se cometer um engano de contratação, despeça-o logo. Pode ser que você perca uns poucos que demandam um tempo maior para amadurecer, mas manter pessoas que não estão contribuindo com sua parte irá prejudicar os demais membros do staff que cobrem o relaxado e resolvem sua incompetência. Em muitas empresas é bastante difícil demitir alguém, especialmente após o período de experiência ter terminado. Certifique-se de que as avaliações iniciais foram feitas de maneira séria. Mais tarde, talvez você tenha de coletar dados revelando a incompetência desse funcionário, dar avisos prévios, definir objetivos de desempenho e assim por diante.

Adequação de atitudes

Os administradores de sistemas normalmente se esquecem de que são provedores de serviços e de que seus usuários são seus clientes. Muitos administradores de sistemas guardam secretamente a opinião de que os sistemas são deles para ficarem mexendo e de que os usuários são um estorvo lamentável.

Os administradores de sistemas são respeitados hoje em dia, tratados como profissionais talentosos e são bem pagos. No passado eram vistos como zeladores eletrônicos várias castas abaixo dos desenvolvedores e engenheiros de sistema. Mas a sua efetividade (corrija o compilador para um engenheiro e ele está corrigido para muitos mais) mudou seu status na maioria das empresas. Eles ainda têm suas premiações para o usuário mais estúpido e outros tipos de brincadeiras inofensivas, mas são muito mais respeitados do que há 10 anos. Os gerentes se dariam bem pedindo a opinião do staff de administração do sistema em relação a possíveis candidatos a promoção; a interação de um usuário com o pessoal de administração do sistema normalmente revela as capacidades de criatividade, independência e resolução de problemas deste usuário.

Algumas das qualidades de um bom administrador de sistemas são contraditórias. Um administrador de sistemas tem de ser impetuoso o bastante para tentar soluções inovadoras quando estiver preso a um problema, mas também tem de ser cauteloso o bastante para não tentar nada realmente destrutivo. Tanto as habilidades interpessoais quanto as de resolução de problemas são importantes, embora pareçam estar em vértices opostos entre os muitos administradores de sistema que conhecemos. Um dos nossos revisores sugeriu que um "administrador de sistemas sociável" era um oximoro.

Os sistemas da Microsoft "sem necessidade de administradores" jamais irão substituir os administradores de sistemas — alguém ainda terá de acionar a tecla <Return> ou clicar OK e assumir a responsabilidade em caso de problemas!

Guerras entre operadores

Administradores de sistemas novos normalmente se tornam vítimas do que chamamos de "guerras entre operadores" nas quais usuários mais experientes fazem **ls** um nome alternativo ("alias") para **logout** nos ambientes dos novatos, enviam e-mail-bombas a eles e geralmente os atormentam e exploram a sua inexperiência. Administradores de sistemas que se esquecem de fazer o logout são vítimas especialmente tentadoras.

Este fenômeno é mais prevalecente em universidades do que em instalações comerciais, mas provavelmente acontece em todo o lado em algum grau. Embora pareça divertido, estas podem facilmente ir longe demais e devem ser estritamente desencorajadas. Já há bastante surpresas reais para um administrador de sistemas novo sem adicionar mais, mesmo de brincadeirinha.

Refinamento iterativo

Um administrador de sistemas normalmente pensará que um problema foi resolvido, apenas para receber vários outros relatos de problemas conforme a tarefa é lentamente feita, completa e corretamente. Este processo pode ocor-

Snow666

rer, pois o usuário que relatou o problema primeiro não o descreveu claramente ou sugeriu a solução errada. Da mesma forma, isso pode ocorrer porque o administrador de sistemas não testou a solução cuidadosamente.

Algumas reclamações comuns são:

- Páginas de manual e documentação não instaladas para um software novo.
- Software que não foi instalado em todos os pontos devidos.
- Software que se revela como de propriedade do administrador de sistemas ou instalado com permissões incorretas.

Testar é chato, mas um administrador de sistemas atarefado pode cortar a produtividade pela metade se pular esta fase. Cada relato de problema custa tempo e esforço, tanto por parte dos usuários como do administrador de sistemas. O trabalho não está completo até que todos os pequenos erros operacionais apareçam e sejam corrigidos.

Um problema comum é que um usuário relata que "X não funciona na máquina Y" e que o administrador vai para a máquina Y, experimenta o tal comando X e ele funciona perfeitamente. A resposta para o relato retorna: "funcionou comigo", com um pouco de ironia nas entrelinhas. Se o administrador realmente tentou o comando conforme o usuário informou no relatório (por exemplo, executando **sudo su -** *nomedousuário* antes do comando), ele poderia pegar o problema exato na primeira tentativa. O argumento "**-**" para **su** faz com que o shell resultante use o ambiente do usuário ao qual você está aplicando **su**. Portanto, você pode realmente reproduzir a situação que foi relatada como não estando funcionando.

Os usuários podem ficar irritados quando um problema não é solucionado completamente na primeira tentativa. Tente definir as expectativas deles apropriadamente. Muitas vezes vale a pena ter o usuário que relatou o problema ao seu lado para tentar solucioná-lo, especialmente se o problema for referente a um pacote de software com o qual você não está familiarizado. Você obterá informações adicionais e será menos provável que o usuário o encare como um adversário.

29.8 Histórias de Guerra e Ética

Por que ética? O trabalho de um administrador de sistemas envolve um nível significativo de confiança, desde respeitar a privacidade do usuário a proteger os segredos da empresa. Relacionamentos de confiança devem ser estabelecidos tanto com seus usuários como com seus superiores. Sem essa confiança, é muito difícil um administrador de sistemas desenvolver bem o seu papel. Suas integridades pessoal e profissional são inestimáveis; proteja-as à medida que ganha confiança de seus usuários, companheiros e superiores.

Esta seção contém alguns dilemas éticos com os quais um administrador de sistemas poderá se deparar. Alguns fatos são experiência própria e outros foram colhidos de fontes externas, talvez de N-ésima mão e talvez não inteiramente em suas formas mais acuradas.

Erro de chefe #1

Um chefe de departamento enviou incorretamente dados pessoais para toda a faculdade em vez de ao comitê-executivo para o qual eram destinados. Ele pediu para um estudante em administração de sistemas que estava trabalhando naquele fim de semana para editar as caixas postais dos membros da faculdade e eliminar a mensagem. O estudante deveria fazer isso? Ou deveria recusar-se? Ele deveria ler a mensagem e decidir por si só se ela realmente era algo sério para se constituir numa invasão de privacidade?

Nesse caso, o administrador de sistemas fez o que lhe foi pedido, porém ele exigiu que esse chefe de departamento enviasse um e-mail para os membros da faculdade explicando o que havia acontecido. Ele também estipulou que houvesse uma testemunha enquanto a mensagem era eliminada das caixas postais alheias e que ela comprovasse que ele, administrador do sistema, não estava vendo mais nada além disso. Essa foi uma boa solução e uma em que tanto o administrador do sistema quanto o chefe do departamento se sentiram à vontade.

Erro de chefe #2

Uma nova secretária num grande fabricante de computadores no centro-oeste era novata em UNIX e e-mail. No final de sua primeira semana, ela enviou uma mensagem ao chefe dizendo de sua satisfação com o novo trabalho e como todo o mundo estava sendo tão solícito. O comando emitido por ela foi algo como o seguinte:

```
% mail boss Eu estou gostando de meu novo trabalho, todo o mundo é tão solícito, obrigada. Trabalhar
  aqui para o senhor vai ser realmente bom …
```

Snow666

O chefe leu o e-mail e respondeu com uma piadinha de mau gosto e com um comentário com apelo sexual sobre o tamanho dos seus seios. Outros colegas responderam sobre a necessidade de colocar um carriage return entre o nome do receptor e a mensagem em si. Parece que "todo o mundo" era um nome alternativo para todos os funcionários.

Algumas horas mais tarde, o administrador de sistemas encarregado (que naquela oportunidade já havia lido tanto a mensagem da secretária como a resposta do chefe) recebeu um telefonema do chefe. O chefe explicou que ele havia cometido um engano (talvez **R** em vez de **r**[10]) e precisava que as cópias da mensagem fossem eliminadas das caixas postais de todo o mundo — sendo que "todo o mundo" eram milhares de funcionários. Não é preciso nem dizer que o administrador de sistemas se recusou a fazê-lo. Uma decisão moral era desnecessária: eliminar uma mensagem de milhares de caixas postais espalhadas pelo mundo é uma tarefa impossível.

Quem despedir?

Um administrador de sistemas novato e um operador trainee descobriram como invadir os computadores dos estudantes do Centro de Computação. Esses hosts eram gerenciados por um outro grupo que era um tanto mal visto por serem muito conservadores. Os novatos queriam deixar um backdoor naqueles computadores. Quando estavam para editar o arquivo **/etc/passwd**, um administrador de sistemas sênior os aconselhou a usar **vipw** em vez de **vi**. Os novatos jamais usaram o backdoor, mas o administrador de sistemas sênior sim e foi pego em flagrante. Quem deve ser demitido?

Nossa resposta seria ou todos os três ou somente o administrador de sistemas sênior, que deveria ter bloqueado a invasão quando ficou sabendo da intenção dos novatos em vez de ajudar e encorajar a instalação do backdoor. Porém, neste caso o administrador de sistemas sênior era muito valioso para ser demitido; conclusão: os dois novatos foram mandados embora.

Vemos isso como uma péssima decisão gerencial. Se os limites do comportamento forem configurados pelo valor de um funcionário em vez de uma política por escrito que seja consistentemente cumprida, a empresa fica vulnerável a litígios (para não dizer nada sobre funcionários descontentes armados com rifles automáticos).

Engenheiros *versus* Departamento de TI

Uma pequena empresa que vendia hardware e software para computadores permitia que seus engenheiros tivessem a senha root em máquinas que não fossem servidores para poderem instalar e desinstalar software ou realizar monitoramento e testes em redes. O **sudo** era disponível, mas o seu emprego não era obrigatório.

Quando a empresa atualizou seus sistemas de Solaris para Linux, o departamento de TI decidiu não distribuir mais a senha root. Um dos engenheiros precisava de acesso root para fazer alguns monitoramentos e testes num sistema Linux. Ele solicitou a senha para o departamento de TI e recebeu a resposta que eles não distribuíam mais a senha root, mas que o departamento de TI faria o que fosse preciso como root. O engenheiro pediu então que o representante do TI logasse na máquina dele e fizesse uma correção. Enquanto o administrador de sistemas estava logado no computador do engenheiro, ele executou um **su** para root para fazer as alterações necessárias. O engenheiro usou então um analisador numa máquina Solaris próxima e obteve a senha root, que ele passou a usar daí para a frente. A situação jamais foi mencionada ao grupo de TI.

Há várias questões envolvidas neste caso. O engenheiro precisava da senha root ou **sudo** para fazer o seu trabalho. Não há nenhuma política por escrito referente à senha root ou análise na rede local. O administrador de sistemas jamais deveria ter confiado na rede local do engenheiro e digitado claramente a senha root via essa rede.

Os problemas jamais foram solucionados. O grupo de TI jamais descobriu que os engenheiros haviam seqüestrado a senha root e que a estavam usando regularmente. Os engenheiros jamais fizeram mau uso desta. Mais motivo ainda para os administradores de sistemas usarem **ssh** consistentemente (mesmo em conexões locais) e de revisarem seus arquivos de log!

Intruso Joe

Joe, um novo administrador de sistemas num grande fabricante de computadores estava cegamente apaixonado pela recepcionista e a convidou para sair. Ela sempre saía com novos funcionários pelo menos uma vez, para mostrar-lhes as redondezas e desejar a eles boas-vindas. Joe convidou-a novamente, mas ela recusou. Cerca de uma semana depois, ela mencionou a um dos administradores de sistemas seniores que sua máquina sempre informava que ela tinha mensagens novas quando na verdade não tinha. Hmmm. O administrador de sistemas sênior verificou os arquivos de log e descobriu que ele estava lendo as mensagens da recepcionista. O que ele deveria fazer?

10. Em alguns clientes de e-mail, **R** significa responder para todos os destinatários da mensagem original e **r** significa responder apenas ao remetente (N.R.T.).

Snow666

- Despedir Joe?
- Repreendê-lo energicamente?
- Conversar com ele de forma mais amena?
- Nada?

A resposta correta é, na verdade, uma pergunta: existe uma política por escrito que diga "Não leia os e-mails dos outros?". A resposta era não. A gerência optou por uma repreensão enérgica. Algumas semanas depois, o fato se repetiu. Dessa vez parecia ser uma pessoa diferente (Tom) que estava lendo os e-mails da recepcionista. O administrador de sistemas sênior chamou Tom para sua sala e lhe apresentou as evidências. Mas Tom estava num jogo de basquete no horário do acesso indicado pelos arquivos de log.

Após investigação adicional e fazer o rastreio por várias máquinas, o administrador de sistemas descobriu que Joe era o verdadeiro culpado mais uma vez. Meia hora depois, Joe era demitido e seus pertences guardados num armário para retirada.

Uma política que permite um alerta é uma licença para roubar até ser pego.

Convites de casamento

Um administrador de sistemas que estava se casando e não havia terminado todos os preparativos para o casamento, deu ao seu padrinho (um administrador de sistemas de uma outra empresa) a chave de seu escritório e a senha root para sua estação de trabalho. O amigo faria, à última hora, cartões de reserva de mesa para deixar na recepção. Esse incidente violava diversas políticas da empresa em que trabalhava e foi notado por outros administradores, pois a prática comum era a de usar o comando **sudo**, e não fazer o login como root ou usando **su**.

A senha root era a mesma em todas as máquinas, portanto o visitante tinha, na verdade, a senha e o acesso físico a toda a instalação. Mas nenhum dano foi causado.

As circunstâncias eram especiais, mas a política escrita havia sido violada. O funcionário era um valioso membro da equipe. O que fazer? Ele foi despedido, de certo modo relutantemente, por justa causa; ele entrou com uma ação e perdeu.

Imagens GIF pornográficas

Um amigo, estudante do colegial, veio visitar o laboratório de computação durante o verão. O estudante mostrou ao seu amigo como visualizar arquivos GIF e indicou a ele alguns lugares "interessantes" para visitar. Ele instalou o amigo na última estação de trabalho no fundo da sala e depois trabalhou em sua tarefa. Ao terminarem, eles saíram.

Tempo depois (dias, provavelmente) o reitor da Engenharia, acompanhado pelo treinador do time de basquete, estava mostrando o campus a uma promissora caloura do Texas (uma mulher). O reitor tinha uma chave para os laboratórios e, portanto, em vez de entrar como os alunos tradicionalmente fazem, através de um cartão de acesso, ele entrou pelos fundos do laboratório usando sua chave.

A primeira estação de trabalho que eles viram era justamente aquela na qual o nosso amigo havia visualizado os GIFs. E graças à maravilha dos protetores de tela, quando o mouse foi movimentado, surgiu sobre uma foto de sexo explícito. Não precisa nem dizer que o reitor e o treinador ficaram furiosos; a estudante não achava que aquilo fosse um grande problema. O reitor ordenou que todos os arquivos GIF fossem eliminados dos computadores da universidade e que os estudantes que haviam feito aquilo deveriam ser expulsos da escola.

Nosso acordo de políticas a serem cumpridas, que o estudante havia assinado, dizia que não deveriam ser exibidas na tela imagens que ofendessem outras pessoas. O final da história foi que o estudante perdeu seu login por um semestre. O acordo foi revisto pelos advogados (que defenderam a posição do departamento de Ciências da Computação, não a do reitor) e todo o incidente foi tratado dentro do departamento. Pedimos desculpas à caloura.

Migração de dados

Uma pequena empresa no Colorado usava uma firma de serviços local para suporte de hardware e de software. Um dia, o administrador de sistemas estava trocando um disco cuja superfície estava ruim. A empresa de manutenção não somente havia fornecido o disco para reposição, como também um grande disco para testes de maneira que a transferência pudesse ser feita sem ter de ir para fita de backup e depois retornar para o disco novo novamente. O administrador de sistemas instalou o disco de reposição e o disco para cópia dos dados originais e reiniciou o sistema.

Ele ficou surpreso quando a estação de trabalho foi inicializada do disco de reposição e reclamou que o relógio estava 297 dias atrasado. Ele deveria limpar o disco imediatamente? Ele deveria primeiro analisar os dados? Ou simplesmente devolver o disco para a empresa de manutenção? O seu primeiro instinto foi o de limpar o disco sem olhar

www.snow666.blogspot.com
Snow666

o seu conteúdo, mas após alguma reflexão achou melhor dar uma olhada em seu conteúdo para que a empresa de manutenção pudessse determinar como ele acabou indo parar lá com dados dentro dele.

Uma rápida varredura no arquivo **passwd** revelou que o disco havia pertencido anteriormente a essa mesma empresa. Ele continha não somente a partição root com senhas criptografadas, como também os bancos de dados de desenvolvimento da empresa e seus novos produtos. Quando perguntada, a firma de manutenção admitiu que a maneira pela qual eles testavam os discos era copiar dados de um para outro, independentemente dos dados que pudessem estar no disco original.

Esse incidente ilustra um problema que é óbvio depois de o fato ter acontecido e também difícil de ser solucionado. De quem é a responsabilidade dos dados num disco com problemas? Ele não pode ser sempre limpo antes de ser restituído para reparo. Os fornecedores de serviços (e, provavelmente, mais importante ainda, os revendedores de periféricos) não vêem necessariamente seus dados como valiosos; eles vêem apenas um disco quebrado ou em vias de.

Como administrador de sistemas, estamos acostumados a proteger nossas fitas de backup. Os discos com defeito são esquecidos e isto não deveria acontecer. Sempre que possível, um disco com dados valiosos deveria ser limpo (uma formatação de baixo nível e uma verificação devem fazer isto) antes de ser mandado para conserto ou comercializado. Se ele estiver muito danificado para ser reformatado, certifique-se de seu fornecedor de serviços estar ciente de que ele contém dados confidenciais que você gostaria que fossem apagados. Considere a possibilidade de colocar cláusulas no contrato de manutenção referentes aos dados da empresa.

Provavelmente vale a pena perguntar à empresa de manutenção sobre a política deles referente aos dados do cliente. Quando admitem que não possuem tal política, haja como se estivesse surpreso e chocado.

Instalações de alta segurança de órgãos governamentais norte-americanos (instalações de defesa, em particular) algumas vezes proíbem terminantemente a saída de qualquer computador da instalação. Se ele quebrar, eles têm de comprar um novo. Pode parecer paranóico, mas como a próxima história ilustra, esta paranóia não é sem fundamentos. (Esta política se aplica até mesmo a componentes como placas de CPU que normalmente não reteriam dados.)

Bush tem de morrer!

Um estudante deixou o computador logado numa máquina do laboratório de computação enquanto ia à sala de seu professor-assistente para pegar um documento. Enquanto ele saiu, alguém escreveu um e-mail para president@whitehouse.gov fazendo ameaças contra o presidente. O Serviço Secreto ligou na manhã seguinte.

O estudante era um estrangeiro que havia servido nas milícias de seu país de origem como um especialista em criptografia. Ele também negligenciou o fato de mencionar aos administradores do sistema local que havia recebido uma confirmação da Casa Branca de um e-mail que ele não havia enviado. As coisas não iam bem.

Os administradores de sistema gastaram o fim de semana coletando arquivos de log e registros de acesso via cartão para determinar o que havia acontecido. Felizmente, os arquivos de log forneciam evidências circunstanciais para convencer o Serviço Secreto de que o estudante provavelmente havia sido vítima de uma brincadeira.

O arquivo de histórico de comandos do estudante (**~/.history**, que incluía horários) confirmava que ele era um usuário regular do **pine** e do **vi**. Mas a mensagem suspeita tinha sido enviada com **emacs**, com um período razoável de inatividade antes e depois do evento. A maioria dos usuários adota tenazmente um editor e um programa de e-mail, de modo que a discrepância era altamente sugestiva de uma conta comprometida.

Como aconteceu, ameaças ao presidente dos Estados Unidos é um crime. Muito embora o estudante estrangeiro tivesse sido absolvido, a investigação do Serviço Secreto prosseguiu. O evento ocorreu uma segunda vez. Novamente foi uma fraude, mas os arquivos de log nos davam informações suficientes para identificar quem pensávamos que estava enviando as mensagens. O Serviço Secreto jamais nos disse se eles processaram a pessoa cujo nome nós demos a eles.

Agora recomendamos que os estudantes usem **xlock**[11] ao deixarem seus terminais sem suas presenças. Nós modificamos **xlock** para fazer com que o usuário saísse automaticamente do sistema após um determinado período de inatividade para que os estudantes não prendam máquinas à toa.

Histórias de segunda mão do World Trade Center

Ouvimos algumas histórias instrutivas relacionadas ao ataque de 11 de setembro de 2001 ao World Trade Center. Infelizmente, os administradores de sistemas que "possuíam" essas histórias foram incapazes de obter a permissão de seus gerentes para compartilhá-las. Em vez de desistir, incluímos aqui algumas histórias de segunda e terceira

11. **xlock** é um programa salva-telas bloqueador para X Windows que requer que você digite a sua senha ou a senha root para desbloquear a tela.

Snow666

mão que talvez não sejam 100% verdadeiras. Elas certamente não são a palavra final, mas de qualquer maneira são interessantes.

Um ISP direcionou todas as suas chamadas e tráfego de rede por meio de uma instalação localizada em um dos menores prédios do complexo do World Trade Center. O prédio do ISP sobreviveu ao ataque e a instalação continuou a operar usando geradores de emergência (embora tivesse tido problemas de dissipar calor, já que o sistema de ar-condicionado do prédio estava inoperante). Infelizmente, o pessoal do ISP estava impossibilitado de adentrar o prédio para reabastecer os tanques de combustível dos geradores. Moral da história: certifique-se de ter capacidade de gerador suficiente para alguns dias para situações de emergência prolongada.

Uma outra empresa localizada em uma das torres do WTC costumava manter sua instalação para recuperação em situações emergenciais na outra torre. Felizmente, a empresa repensou sua definição de "off-site" antes do ataque. Uma terceira empresa foi capaz de recuperar todos os seus dados críticos que eram replicados em vários dispositivos em diversos locais diferentes, espelhados sincronicamente via fibra ótica. Entretanto, essa empresa em particular perdeu uma série de documentos em papel — nem tudo estava na forma computadorizada.

O site Web da CNN experimentou um tráfego extraordinário no dia 11 de setembro que desafiou as capacidades de medição dos monitores de carga da CNN. O sistema telefônico em Nova Iorque ficou sobrecarregado e os canais de TV que costumavam transmitir de antenas sobre o WTC saíram do ar. Todo o mundo pulou para a Internet e castigaram o site cnn.com. Inicialmente o site ficou impotente diante da sobrecarga. Após tentar contornar o problema deslocando servidores, a CNN acabou reduzindo a complexidade de sua homepage para uma única imagem e texto puro sem links. Após algumas horas, o site estava recuperado e conseguindo atender à demanda.

29.9 Documentatação Local

A documentação normalmente é relegada a um segundo plano na lista de prioridades em benefício do "trabalho de verdade". É relativamente fácil protelar a documentação, pois no momento em que deveria estar redigindo-a, você se lembra de como fazer a tarefa em questão e não tem necessidade de uma mísera folha de papel. Qualquer grupo de administração que envolva estudantes provavelmente tem sérios problemas de documentação.

A documentação local serve para muitas coisas. Alguma vez você já teve de entrar numa sala de máquinas para reinicializar um servidor, somente para se deparar com racks e racks de hardware, todos parecidos entre si, todos diferentes e todos sem nenhuma identificação? Ou teve de instalar um componente de hardware com o qual já lidou no passado, mas tudo o que você consegue se lembrar agora a respeito da tarefa é que ela era difícil de ser feita? Ou ter passado horas e horas localizando uma máquina nova para se ajustar ao seu ambiente apenas para descobrir que você se esqueceu de uma série de etapas cruciais?

A documentação local deve ser mantida num ponto bem definido, talvez em **/usr/local/doc**. Certos documentos são mais apropriados na forma de uma brochura ou como uma indicação pendurada num dispositivo de hardware.

Os servidores principais devem ser rotulados com o nome de host, instruções para inicialização, arquitetura e quaisquer instruções especiais necessárias para reinicializar a máquina. O nome de host deve ser legível de qualquer ponto da sala. Certifique-se de manter uma cópia das informações contidas naquelas pequenas etiquetas colantes junto com os seus registros centrais ou dados de inventário.

Também associe o nome de host com outros dispositivos de hardware relacionados a cada máquina: unidades de disco, modems, impressoras, unidades de fita etc. Se o host for um membro importante (por exemplo, um servidor principal ou um roteador crucial), inclua a localização de seu disjuntor. Se for necessário um disquete ou cartão de memória flash para inicialização, aponte para sua localização. Os servidores de arquivos principais devem ter informações sobre nomes de dispositivos de disco, tabelas de partição, pontos de montagem e as localizações de superblocos de backup prontamente disponíveis. Coloque as informações para as próprias unidades de fita ou as armazene numa localização bem conhecida na sala de máquinas.

As unidades de fita devem ser identificadas com os arquivos de dispositivos e comandos necessários para acessá-las. É uma boa idéia listar o tipo de fitas que a unidade requer, o local mais próximo para adquiri-las e até mesmo o seu preço aproximado.

As impressoras devem ser rotuladas com seus nomes, instruções de impressão breves e os hosts dos quais elas dependem. Normalmente as impressoras vêm com as interfaces de rede e são membros completos da rede, mas elas podem depender de um host Linux para spooling e configuração.

A interligação de redes deve ser documentada cuidadosamente. Rotule todos os cabos, identifique painéis de ligação e tomadas e marque os dispositivos de rede. Sempre facilite as condições para que o técnico responsável pelas ligações consiga manter a documentação atualizada; mantenha lápis e formulários pendurados na parede do gabinete de fiação para facilitar as anotações de que um cabo foi transferido de um dispositivo para outro. Posteriormente, você deve transferir esses dados para armazenamento on-line.

www.snow666.blogspot.com
Snow666

Um arquivo **diário** associado a cada máquina e que documente os principais eventos de sua vida (atualizações, reparos no hardware, principais instalações de software e quebras, por exemplo) fornece um lugar centralizado para rever o histórico e o estado de uma máquina. Você pode apontar um nome alternativo de e-mail para o arquivo de maneira que o diário possa ser enviado via e-mail com cópia para todos os administradores de sistemas. Essa talvez seja a maneira mais fácil e menos organizada de manter registros, porém sua simplicidade a torna mais fácil de ser cumprida.

É interessante preparar um documento impresso que você possa entregar a usuários novos. Ele deveria documentar personalizações específicas, procedimentos para relato de problemas, os nomes e as localizações de impressoras, seus cronogramas de backup e de desligamento programado e assim por diante. Esse tipo de documento pode lhe poupar uma quantidade enorme de tempo de serviços de usuários e de administradores de sistemas. Você também deveria tornar essas informações disponíveis na Web. Um documento impresso será mais provavelmente lido pelos novos usuários, porém uma página Web é mais fácil de ser acessada nos momentos em que surgirem dúvidas. Implemente ambos os procedimentos.

Além de documentar seu ambiente computacional local, talvez seja interessante preparar algum material introdutório ao Linux. Tal material é essencial num ambiente universitário no qual a comunidade de usuários é transiente e muitas vezes não conhece Linux. Temos "colas" impressas de uma página sobre o editor **vi**, e-mail, procedimentos para entrada e saída do sistema, sobre o ambiente X Windows e a utilização das páginas de manuais.

29.10 Compras

Em muitas instalações, a equipe de administração de sistemas e a equipe de compras são totalmente separados. Isso é ruim.

Os administradores de sistemas precisam saber sobre qualquer hardware novo que estiver sendo comprado para verificar se ele se encaixa na infra-estrutura atual e tem condições de ser suportado. Eles também precisam estar aptos a influenciar as especificações que vão nas solicitações de compras. Os administradores de sistemas normalmente fornecem boas informações sobre a competência dos fornecedores (especialmente revendedores) e a confiabilidade de certos tipos de equipamentos.

A participação de um administrador de sistemas é especialmente valiosa em organizações que por padrão têm de comprar a oferta mais baixa (por exemplo, instituições governamentais e universidades estaduais). A maioria dos sistemas de compras permite que especifiquemos critérios de avaliação. Certifique-se de incluir cláusulas de escape como "tem de ser compatível com o ambiente existente" ou "tem de ser capaz de rodar bem o pacote XYZ".

O custo incremental de adicionar uma estação de trabalho não é constante. Ela é a 60ª ou a primeira deste tipo? Ela possui disco local suficiente para os sistemas de arquivos? Ela possui memória suficiente para executar os aplicativos pesados de hoje em dia? Há uma porta de rede reserva para conexão? Ela ficará numa área do prédio que é acessível à rede? Trata-se de um sistema operacional completamente novo?

Perguntas como essas tendem a enfatizar uma questão mais fundamental: você fica estagnado e compra equipamento do fornecedor atual ou você tenta a novidade mais incrementada de uma empresa que poderá ter estrondoso sucesso ou quem sabe ter falido após um ano? A natureza de sua empresa deve responder a essa pergunta. Não é um simples sim ou não; normalmente é necessário fazer uma comparação complexa entre o mais recente e arrojado equipamento e as máquinas com as quais já está acostumado e compreende.

Se lhe for permitido negociar com fornecedores (oficialmente ou de alguma outra maneira) em geral você pode se dar muito melhor do que com o seu próprio departamento de compras. Não se acanhe em relação a fazer cotações com outros fornecedores de equipamentos equivalentes ou de inflacionar o valor esperado das compras para o ano seguinte. Afinal de contas, o pessoal de vendas inflacionou o valor de seus produtos. Estar apto a obter um pedido de compras de forma bem ágil é útil na hora de gastar aquele fundo de caixa em épocas como final do trimestre ou ano fiscal. Ao fazer pedidos de compra na última semana de um período contábil de uma empresa, muitas vezes se pode obter um desconto adicional razoável para fazer com que a cota de um departamento pareça melhor. Um outro período de barganha é logo antes ou logo depois da introdução de novos modelos; os fornecedores querem reduzir seus estoques de produtos mais antigos. Esse tipo de golpe é comum em universidades e pode ou não ser apropriado em empresas e órgãos governamentais. Mas é bom.

29.11 Afastando Hardware de Serviço

Aposentar um computador usado pode ser uma tarefa difícil. Usuários teimosos não o deixarão ir; o "desmamamento" desses usuários requer que eles aprendam a usar um novo sistema ou façam a conversão para novos aplicativos. Mas já que o hardware Intel/PC barato se tornou a arquitetura-base padrão, aposentar máquinas antigas simples-

Snow666

mente significa adicionar mais memórias ou substituir a placa de CPU. A interface com o usuário permanece a mesma; tudo simplesmente fica mais rápido.

Um problema relacionado com universidades envolve doações de empresas interessadas em obter reduções no imposto de renda. Normalmente a resposta correta é: "Não, obrigado, não precisamos de 2.000 unidades de fita de nove trilhas e racks para armazená-las". Há muitos anos uma universidade em Budapeste recebeu como doação um mainframe IBM. Em vez de dizer não e comprar PCs rápidos, eles dedicaram todo o orçamento para transporte e fiação elétrica no primeiro ano e manutenção nos anos seguintes. Resultado é que não valeu a pena. Por outro lado, muitas universidades estabelecem fortes vínculos com a indústria de informática local e obtêm uma série de hardware valioso como doação — são bons modelos do ano passado. Considere o custo total de propriedade e saiba quando dizer: "Não, obrigado".

Se o seu caso for o de uma empresa e tiver máquinas sobrando, considere a possibilidade de doá-las para escolas ou faculdades de sua região; elas normalmente não têm verba suficiente para manterem seus laboratórios atualizados e podem aceitar com muito prazer os seus PCs mais antigos.

29.12 Organizações, Conferências e Outros Recursos

Vários grupos de suporte UNIX e Linux — tanto genéricos quanto específicos de fabricantes — existem para ajudá-lo a estabelecer uma rede de contatos que usam o mesmo software que o seu. A Tabela 29.1 apresenta uma lista resumida de organizações. Há uma quantidade enorme de grupos nacionais e regionais que não são apresentados nesta tabela.

Tabela 29.1 Organizações patrocinadoras do Linux e do UNIX.

Nome	URL	O que ela é
LPI	www.lpi.org	Linux Professional Institute
LI	www.li.org	Linux International
USENIX	www.usenix.org	Grupo de usuários UNIX; bem técnico
SAGE	www.sage.org	System Administrators Guild associada ao USENIX; sedia a conferência LISA todos os anos
SANS	www.sans.org	Patrocina conferências sobre segurança e administrador de sistemas ; menos técnica
AUUG	www.auug.org.au	Australian UNIX Users Group, cobre aspectos tanto técnicos quanto de gerenciamento
SAGE-AU	www.sage-au.org.au	SAGE australiana, sedia anualmente conferências na Austrália.

Algumas organizações realizam conferências sobre tópicos relacionados com o Linux embora a maioria das conferências seja mais abrangente do que simplesmente Linux ou UNIX e incluem também eventos para Windows. A Linux International e o Linux Professional Institute promovem o Linux de várias maneiras — a LI através de esforços de marketing para trazer o Linux para a comunidade comercial (e financiar algumas das iniciativas para o desenvolvimento de fonte aberto) e o LPI através de seu programa de certificação de administração de sistemas Linux.

O site Web da LI possui bons subsídios para convencer gerentes de sistemas relutantes ao emprego de sistemas operacionais de fonte aberto. Eles patrocinam um estande Linux em diversas feiras, entre as quais a LinuxWorld, a Linux Expo e a CeBit. A LI também está conduzindo um projeto de internacionalização para trazer os conjuntos de caracteres de todas as línguas para o Linux. O programa de internato nas férias de verão reúne estudantes com desenvolvedores Linux para dar aos estudantes experiência prática na criação de programas de alta qualidade para ambientes de produção.

A USENIX, uma organização de usuários de Linux, UNIX e outros sistemas operacionais de fonte aberto, sedia uma conferência geral e vários workshops e conferências especializadas (menores) a cada ano. A conferência tem um espaço paralelo dedicado a sistemas abertos que apresenta desenvolvimento de sistemas operacionais em andamento nas comunidades Linux e BSD. Em 2000 e 2001, a USENIX aliou forças com a comunidade Linux para patrocinar a ALS (Atlanta/Annual Linux Showcase), uma conferência de uma semana de duração dedicada ao Linux. O grande evento para administrador de sistemas é a conferência USENIX LISA (Large Installation System Administration) que ocorre no terceiro trimestre de cada ano. As feiras normalmente são associadas a estas conferências.