



- LPI 102 -

Administração de Redes Linux

Há mais de 17 anos, a **GREEN Treinamento** vem atuando exclusivamente com treinamento e possui toda a infra-estrutura para oferecer-lhe a melhor experiência no aprendizado de informática.

Já treinamos mais de **130.000** alunos de todas as procedências: grandes empresas, particulares, microempresários, estudantes, etc.

Somos um Centro de Treinamento Oficial certificado pelos três principais fabricantes da indústria de softwares: nós somos **CPLS da Microsoft (antigo CTEC)**, **ECIS da IBM** e **Training Provider da Mandriva Conectiva**, ou seja, atendemos aos padrões internacionais de qualidade destas empresas, contando com profissionais certificados e informações técnicas e comerciais sempre atualizadas.

Somos **Centro de Testes da Pearson VUE – Virtual University Enterprises**. Podemos aplicar testes para diversas certificações, entre as quais: Novell, Ericsson, Microsoft, Lucent, LPI (Linux), Informix, CompTIA, Certified Internet Webmaster, Lotus, Avaya, Allaire, Siebel, BROCADE, Cisco, RSA Security, NASD, Check Point, Macromedia, GNU e Intershop Communications.

Possuímos modernos laboratórios equipados com computadores de última geração. Existem também unidades móveis onde instalamos os equipamentos no cliente para ministrar cursos "IN-HOUSE".

"Oferecer a melhor capacitação profissional é o nosso objetivo"



Learning Solutions



GREEN TREINAMENTO

Informações: (11) 3253-5299

Av. Paulista, 326 - 12º Andar

Bela Vista - São Paulo - SP

Metrô Brigadeiro

(Saída Carlos Sampaio)

<http://www.green.com.br>

cursos@green.com.br

Estacionamentos Conveniados

Desconto válido somente com carimbo da recepção no ticket do estacionamento.

Créditos

Marcos Sungaila
autor

Copyright Marcos Sungaila

TODOS OS DIREITOS RESERVADOS. É proibida a reprodução parcial ou total desta obra, por qualquer meio, seja este gráfico, fotográfico ou eletrônico entre outros, bem como a inclusão deste trabalho em qualquer sistema de arquivo de processamento de dados (disquetes, cd's, dvd's, etc), sem prévia autorização por escrito de Marcos Sungaila. Tais vedações se aplicam também à supressão de informações através da alteração das características originais deste trabalho, incluindo alterações de diagramação e de características gráficas da obra.

A violação de direitos autorais é crime, nos termos da lei 9.610/98, e conforme estabelecido pelo artigo 184 do Código Penal. Aplica-se a esta obra e em relação aos nomes empresariais, marcas mistas, figurativas ou nominativas contidas na mesma, as disposições legais da lei 9.279/96.

A reprodução deste material sem autorização escrita de Marcos Sungaila é crime, de acordo com a legislação em vigor. Este material não pode ser fotocopiado.

Se você verificar que esta obra foi fotocopiada ou reproduzida de qualquer outra forma, entre em contato com o autor pelo e-mail marcos@savant.com.br.

Copyright 2006, Marcos Sungaila

Índice

Fundamentos de rede.....	4
O que são redes de computadores?.....	4
TCP/IP.....	4
Endereços IP.....	5
Configuração de rede.....	7
Acesso via PPP (discado ou adsl).....	10
Ferramentas básicas de rede.....	12
ping – echo request.....	12
traceroute – caminho percorrido pelo pacote.....	12
dig – informações sobre nome de host.....	13
whois – localizando informações sobre domínios.....	13
Serviços de rede.....	14
Serviços sob demanda.....	14
inetd.....	14
xinetd.....	15
Servidor de email.....	16
Instalando o sendmail.....	16
Redirecionamentos de email – aliases.....	17
Permitindo o envio de mensagens pelo servidor.....	18
Configurando o domínio.....	18
Iniciando o sendmail.....	18
Testando o servidor.....	19
Sendmail – Referências.....	19
Servidor web.....	20
Instalando o apache.....	20
Configuração básica do apache.....	20
Acesso autenticado.....	22
Servidor de arquivos NFS.....	23
Acessando um servidor NFS.....	23
Compartilhando dados com NFS.....	24
NFS – Referências.....	24
Servidor de arquivos SAMBA.....	25
Instalando o Samba.....	25
Configuração básica do Samba.....	26
Servidor DNS.....	28
Tipos de configuração do BIND.....	28
Arquivos de configuração do bind.....	28
Configurando o BIND como um servidor cache com forward.....	29
Configurando o BIND como servidor autoritário de domínio	31
Criando um servidor DNS Slave.....	33
DNS – Referências.....	34
DHCP – Dynamic Hosts Configuration Protocol.....	35
Instalando e configurando o DHCP Server.....	35
Clientes dhcp.....	37
dhpcd – dhcp client daemon.....	37

dhclient.....	38
pump.....	38
DHCP – Referências.....	38
Acesso remoto.....	40
SSH – Secure Shell.....	40
Fazendo uma conexão ssh pela primeira vez.....	41
Copiando arquivos entre máquinas.....	42
Usando chaves RSA/DSA com ssh.....	42
SSH – Referências.....	43
Segurança básica.....	44
O que é segurança.....	44
Segurança Física.....	44
Segurança de software.....	45
Ferramentas de segurança.....	45
Password Shadow suite – Senhas.....	46
tcp wrapper.....	46
Firewall básico.....	47
Usando o iptables,.....	47
ipfwadm e ipchains.....	48
Firewall – Referências.....	48
Segurança local – desativação de serviços.....	49
Monitorando os logs do sistema.....	49
Apêndice A – Configurações adicionais do Apache.....	50

Fundamentos de rede

Podemos dizer que um dos grandes avanços tecnológicos do século 20 foi a criação das redes de computadores e principalmente o surgimento da Internet e o seu uso como meio de comunicação.

Desde as primeiras implementações na década 60 até hoje várias formas de conectar equipamentos em rede surgiram e são utilizadas, entre elas está a comunicação via protocolos tcp/ip, padrão na Internet. Devido a isto e ao fato de o Linux ter estes protocolos com padrão de comunicação é que serão abordados durante nosso curso.

Apesar de existirem vários outros protocolos em uso na Internet, o padrão tcp/ip é, sem sombra de dúvida, o mais difundido e utilizado atualmente.

O que são redes de computadores?

São um grupo de computadores interligados por um meio de comunicação de dados normalmente compostas por hosts (servidores, estações, impressoras, switches), gateways, routers e vários outros componentes.

Um host é um equipamento ligado à rede e que possui um endereço próprio e não é capaz de encaminhar pacotes a outras redes.

Um gateway tem como característica estar conectado a mais de uma rede física e é capaz de encaminhar os pacotes de um segmento (rede) para outro.

Um gateway normalmente é um equipamento especializado ou um host que é capaz de converter protocolos, entre TCP/IP e IPX/SPX por exemplo ou SNA. Este tipo de recurso ocorre em níveis mais altos na pilha de protocolos.

Uma rede simples é basicamente formada por um Servidor e suas estações ou clientes. O Servidor centraliza todos os recursos e informações permitindo o compartilhamento de dados e recursos disponíveis nele. As estações ou clientes são os computadores que se encontram conectados ao Servidor usufruindo dos recursos que ele oferece.

As redes que utilizam TCP/IP rodam sobre uma estrutura básica como:

- Ethernet
- Comunicação serial (PPP ou SLIP)
- FDDI
- Token Ring

TCP/IP

A suíte de protocolos TCP/IP foi criada em 1969 como fruto de um projeto da DARPA (Defense Advanced Research Projects Agency). Este projeto tinha como objetivo criar uma rede para integração de pesquisa e troca de informações para os órgãos do governo. Esta rede foi

batizada de ARPANET e, em 1983, foi separada em duas dando origem à MILNET (de cunho militar com a grande maioria dos pontos de acesso da época) e à ARPANET (bem menor e de cunho acadêmico) que deu origem à Internet como a conhecemos hoje. A ARPANET foi formalmente extinta em 1990.

Inicialmente desenvolvido e utilizado no ambiente UNIX, os protocolos TCP/IP estão disponíveis em vários sistemas operacionais incluindo:

- Linux
- Unix
- MacOS
- Novell
- MS Windows® desde WFW3.11
- e muitos outros

Entre os vários protocolos utilizados na Internet e que compõem a suíte TCP/IP temos:

- Internet Protocol (IP)
Utilizado para a identificação de hosts, é o responsável pelo roteamento de pacotes na Internet definindo o caminho a ser utilizado em uma comunicação. Todos os outros protocolos da suíte TCP/IP com exceção de ARP e RARP utilizam o IP para roteamento.
- Transmission Control Protocol (TCP)
Utilizado para o controle da transmissão tem características como fragmentação de dados, controle de fluxo e correção de erro. Devido aos recursos implementados necessita de confirmação de comunicação sendo normalmente conhecido como um protocolo orientado à conexão.
- User Datagram Protocol (UDP)
Protocolo rápido e leve é utilizado para comunicação pontual como consultas a servidores DNS ou transferência de arquivos de forma direta. Por não possuir controle de conexão tem baixo overhead. Sua implementação é pouco confiável, deixando a cargo da aplicação a correção de possíveis erros de comunicação.
- Internet Control Message Protocol (ICMP)
Utilizado para testes de alcance, o ping como é mais conhecido tem características de identificar dados do equipamento remoto como data e hora local. Possui recursos de controle e/ou caráter informativo como o redirecionamento de rotas ou o congestionamento de redes.
- File Transfer Protocol (FTP)
Utilizado para a transferência de arquivos este protocolo de mais alto nível trabalha sobre tcp implementando recursos e controles adicionais.

Endereços IP

Todo equipamento conectado a uma rede TCP/IP deve ter um endereço próprio e único. Os endereços IP podem ser do tipo IPV4 largamente adotados ou IPV6 ainda em fase de disseminação e pouco utilizado na Internet.

- O padrão IPV4 define um endereço de 32 bits (4 octetos) em notação decimal pontuada variando de 0 a 255 em cada octeto.

- Um endereço IP é mapeado para um hardware através dos protocolos ARP/RARP.

Alguns endereços IP não são utilizado na Internet. São reservados:

- 0 – não utilizado
- 10 – utilizado para redes privadas de grande porte (até 16.777.216 ip's)
- 127 – reservada para loopback
- 172.16 a 172.31 – reservada para rede privadas de médio porte (até 1.048.576 ip's)
- 192.168 – utilizada para redes privadas de pequeno porte (até 65.536 ip's)

Endereços IP são conhecidos como endereços lógicos enquanto o endereço físico de um equipamento, como em uma rede Ethernet, é o MAC Address de sua placa.

Inicialmente o endereçamento IP foi criado em Classes (faixas), definidas de acordo com o número de hosts e redes possíveis em cada uma. As classes de endereços IP possuem as seguintes faixas:

CLASSE	DE	ATÉ
A	0.0.0.0	127.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D	224.0.0.0	239.255.255.255
E	240.0.0.0	255.255.255.255

Os endereços IP's foram definidos como sendo do tipo:

- Unicast: comunicação direta host a host
- Multicast: comunicação para um grupo de máquinas
- Broadcast: comunicação para todas as máquinas de um segmento de rede

A configuração das informações de rede no Linux envolvem vários arquivos no diretório /etc, que definem, entre outras coisas, o nome da máquina, resolução dinâmica de nomes (DNS) e resolução estática (hosts). Os arquivos mais comuns são:

ARQUIVO	DESCRIÇÃO
hosts	Mapeamento estático entre ip e nome de host, por exemplo: # IP Hostname Apelido 127.0.0.1 localhost.localdomain localhost 192.168.2.200 instrutor.sala2 instrutor
host.conf	Controla a resolução de nomes de máquinas em sistemas mais antigos
hostname	Nome do equipamento. Em alguns sistema é escrito em

```
# ifconfig eth0 192.168.0.1 netmask 255.255.255.0
```

No Debian, a configuração permanente (que será ativada durante a inicialização do serviço) é colocada no arquivo /etc/network/interfaces. Sua estrutura está mostrada abaixo.

```
auto lo                                Indica o dispositivo de saída
iface lo inet loopback                  Define a interface de preferência para a rota de saída

auto eth0                               Indica o dispositivo de saída
iface eth0 inet static                  Define a interface de preferência para o equipamento;
    address 192.168.0.1                para o equipamento;
    netmask 255.255.255.0              para o equipamento;
    network 192.168.0.0               para o equipamento;
    broadcast 192.168.0.255           para o equipamento;
    gateway 192.168.198.254          para o equipamento;
```

Para que esta configuração seja ativada basta reiniciar nosso serviço de rede.

```
# /etc/init.d/networking restart
```

Esta é uma configuração onde são definidas todas as informações da interface e que possibilitam a qualquer equipamento um acesso completo à rede.



Em distribuições como Mandriva, Fedora, Redhat, Suse e Conectiva a configuração das interfaces é realizada pelos arquivos ifcfg-ethN, localizados no diretório /etc/sysconfig/network-scripts. Sua estrutura básica está lista a seguir:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.198.128
NETMASK=255.255.255.0
TYPE=Ethernet
USERCTL=yes
IPV6INIT=no
PEERDNS=yes
HWADDR=00:0C:29:35:0A:4F
```

Estas distribuições utilizam, ainda, o arquivo /etc/sysconfig/network para indicar o gateway a ser utilizado, apesar desta informação constar do arquivo acima, como abaixo:

```
NETWORKING=yes
HOSTNAME=instrutor.sala2
GATEWAY=192.168.198.2
GATEWAYDEV=eth0
```

A configuração da rota padrão de um equipamento pode ser feita manualmente, através do comando route, o seu funcionamento e suas opções estão listadas abaixo.

Acesso via PPP (discado ou adsl)

O Protocolo para conexões Ponto a Ponto (Point to Point Protocol – PPP) é o padrão utilizado em conexões variadas, entre elas a conexão discada e acesso de banda larga adsl que utiliza o PPP sobre conexões ethernet (pppoe).

As conexões do tipo ppp podem ser utilizadas para:

- conectar um equipamento Linux a um servidor remoto, por exemplo a um provedor de acesso Internet;
- transformar seu equipamento em um servidor de acesso remoto (RAS);
- conectar dois equipamentos Linux para integrar duas redes em localidades diferentes, definindo uma wan.

A configuração do acesso a servidores remotos pode ser feito em modo texto com uma aplicação chamada `chat` que utiliza uma série de comandos pré-definidos para poder se conectar ao equipamento remoto ou através da interface gráfica com o programa `kppp`, por exemplo, que toma conta de todo o processo de discagem, conexão e autenticação.

O processo de conexão em modo é feito com a execução do serviço `pppd`, que ao ler seu arquivo de configuração `/etc/ppp/options` aciona o script de conexão. Um exemplo desta configuração está a seguir:

```

/dev/modem          # dispositivo do seu modem
lock                # garante o acesso exclusivo ao modem
auth               # exige autenticação antes do envio de pacotes
defaultroute       # define a rota padrão após a conexão
modem              # utiliza as linhas de controle pelo modem
115200             # velocidade de comunicação do PC com o modem
crtscs             # controle de dados da porta serial por hardware
noipdefault        # utiliza o ip fornecido pelo provedor
usepeerdns         # utiliza os servidores fornecidos pelo provedor
connect /etc/ppp/conectar # script utilizado para discar e conectar

```

PERSISTENT

não cai a conexão dial-up → conexão mais instável

Para que este serviço consiga discar e autenticar o usuário no provedor, ainda são necessários alguns passos adicionais. O programa que será utilizado para acessar o modem e discar para o provedor é o `chat`. Este comando deve poder interpretar as saídas do modem para poderm decidir que ação tomar. Abaixo temos um arquivo de configuração com as opções mais comuns para conexão onde são testados alguns erros e eventos bem sucedidos.

`/etc/ppp/chat.conf`

```

ABORT BUSY \
ABORT 'NO CARRIER' \
ABORT VOICE \
ABORT 'NO DIALTONE' \
ABORT 'NO DIAL TONE' \
ABORT 'NO ANSWER' \
ABORT DELAYED \
'' ATZ \
OK-AT-OK "ATDTxxxxxxxx" \
CONNECT '' \
login: "usuario" \
ssword: "\qsenha-de-acesso"

```

Na configuração do servidor `pppd` informamos que o comando de conexão será o

/etc/ppp/conectar. Vamos criá-lo com o conteúdo abaixo e em seguida atribuir permissão de execução a ele.

```
#!/bin/bash  
/usr/sbin/chat -v -f /etc/ppp/chat.conf
```

Como finalização, vamos criar um pequeno script como abaixo que finalize o serviço pppd para podermos desconectar nosso acesso e atribua permissão de execução a ele.

```
/usr/local/bin/desconecta
```

```
#!/bin/bash → DESLIGA A CONEXÃO  
/usr/bin/pkill HUP pppd →
```

Com esta estrutura criada, para você se conectar ao provedor bastará iniciar o serviço pppd.

```
# pppd → Conecte internet
```

A característica do pppd no Linux é que ele não para após um número qualquer de tentativas de conexão. Você irá precisar escrever uma script para cancelar o comando e encerrar o caminho de pacotes que ele deve enviar.

Escreva o que quer pacote de rede e finalizando o comando.

Verificando o funcionamento da rede devem envolver os seguintes destinos de ping:

localhost 127.0.0.1 – significa que a configuração básica tcp/ip está correta

192.168.1.1 – é o endereço ip da sua roteadora – representa que a conectividade local está funcionando

192.168.1.100 – é o endereço da sua interface – sua máquina roteadora pode acessar a internet diretamente

www.green.com.br – seu roteador ou seu provedor pode resolver o nome de sites na internet

Tracertute – caminho percorrido pelo pacote

Além de ser possível descobrir por onde um pacote está passando para chegar a determinado destino, pode-se tracar o caminho percorrido pelo comando tracertute que envia pacotes do tipo echo com características específicas.

Exemplo: tracertute mostra pacotes enviados para um determinado site

```
# traceroute www.green.com.br
traceroute to www.green.com.br (200.234.196.78), 30 hops max, 38 byte packets
 1 192.168.198.2 (192.168.198.2) 4.044 ms 0.645 ms 0.333 ms
 2 hm231.locaweb.com.br (200.234.196.78) 27.021 ms 9.601 ms 19.967 ms
```

Atualmente muitos hosts não respondem ao ping (echo request) que é o pacote utilizado para estes testes resultando que o uso do comando traceroute dificilmente funciona de acordo com o esperado.

dig – informações sobre nome de host

Pesquisa por informações sobre nome de hosts ou endereços ip a partir da pesquisa especificada. Pode ser utilizado ainda para encontrar registros especiais de domínio como o endereço do servidor MX (que recebe emails) ou NS (servidor DNS).

Resolvendo nomes:

```
# dig www.green.com.br
```

Encontrando nomes a partir de endereços IP:

```
# dig -x 200.221.2.45
```

whois – localizando informações sobre domínios

Este comando consulta os dados de domínios cadastrados nos órgãos responsáveis como registro.br no caso de domínios nacionais.

Pesquisando os dados de um domínio:

```
# whois green.com.br
```

Serviços de rede

Vários recursos podem ser disponibilizados para a rede por um equipamento, compartilhamento de arquivos, impressão ou transferência de arquivos criando um repositório ou permitir acesso remoto à rede. Estes são os tipos de serviços que serão vistos em nosso curso.

Serviços sob demanda

Todos os recursos listados acima podem ser oferecidos de forma permanente (ligados o tempo todo) ou serem acionados apenas quando necessário. Este acionamento sob demanda é possível através do super servidor `inetd` ou a versão aprimorada `xinetd`.

inetd

Este servidor abre as portas especificadas em seu arquivo de configuração e, quando recebe uma conexão nesta porta, aciona o servidor correspondente. A grande vantagem na disponibilização de serviços desta forma é que o servidor `inetd` usa pouquíssimos recursos do equipamento, permitindo que vários serviços sejam disponibilizados pela mesma máquina uma vez que somente serão acionados quando necessário.

Sua configuração especifica que porta deve ser aberta e como ocorre a conexão (protocolo, espera) e qual é o servidor que deve ser acionado com que usuário quando uma conexão é recebida. Instale o pacote `telnetd` e veja um exemplo para este servidor.

```
# apt-get install telnetd
#
# grep telnet /etc/inetd.conf
telnet stream tcp nowait telnetd.telnetd /usr/sbin/tcpd /usr/sbin/in.telnetd
```

Esta linha de identificação tem a seguinte estrutura:

- serviço
Porta a ser liberada para acesso, de acordo com o cadastro no arquivo `/etc/services`
- socket
Forma de transmissão dos dados, pode ser `stream` para conexões `tcp` ou `dgram` para conexões `udp`
- protocol
Protocolo usado para conexão (`tcp` ou `udp`)
- flags
Utilizado como `wait` e `nowait`. A flag `wait` define que servidor deve disponibilizar conexões imediatamente para que você tenha várias conexões simultâneas com em

um serviço de ftp. A flag `wait` disponibiliza apenas uma conexão ao servidor especificado por vez.

■ usuário

Qual o usuário que deverá acionar o servidor quando uma conexão for recebida.

■ Servidor inet

O servidor que abre as portas e controla a conexão aos serviços é, normalmente, o `tcpd`.

■ argumentos

Normalmente é o servidor a ser acionado com seus parâmetros de ativação.

Ligue o serviço inet e faça um teste de acesso:

```
# /etc/init.d/inetd start
#
# telnet 127.0.0.1
```

xinetd

O servidor `xinetd` foi criado como um substituto mais seguro para o `inetd`. Permite que qualquer usuário inicie um serviço desde que este não utilize portas privilegiadas (abaixo de 1024). Possui uma configuração própria que controla o servidor em `/etc/xinetd.conf` e arquivos adicionais para cada serviço a ser acionado dentro do diretório `/etc/xinet.d`.

```
# Configuração mínima recomendada para o servidor xinetd

defaults
{
    instances          = 25
    per source         = 10
    log_type           = SYSLOG authpriv
    log_on_success     = HOST USERID PID
    log_on_failure     = HOST USERID
}

includedir /etc/xinetd.d
```

Por exemplo, a ativação do servidor `telnetd` feito pelo `xinetd`, exige a criação de um arquivo específico no diretório `/etc/xinetd.d`, como este:

`/etc/xinetd.d/telnet`

```
service telnet
{
    disabled          = no
    socket_type       = stream
    wait              = no
    user              = root
    server            = /usr/sbin/in.telnetd
    log_on_failure    += USERID
}
```

Servidor de email

Os servidores de email se tornarão a parte mais importante do acesso Internet nos dias de hoje. Negócios são fechados por email. Compras são confirmadas e reuniões são agendadas com base em mensagens dos participantes. Os serviços de correio eletrônico envolvem vários componentes diferentes:

- servidor smtp que se encarregar de enviar a mensagem ao destinatário
- servidor pop para baixas as mensagens do provedor para o nosso equipamento
- servidor imap para podermos ver nossas mensagens de qualquer lugar através de um webmail
- programa de correio eletrônico que nos permite ler e criar nossas mensagens

De todos estes componentes é no servidor smtp que reside a maior parte do trabalho para o correto funcionamento de todo este esquema complexo que é o envio e recepção de emails.

Os componentes envolvidos na criação, envio e recepção de mensagens entre servidores e a sua entrega na caixa postal do usuário são:

- Mail User Agent (MUA): o client de email utilizado pelo usuário
- Mail Transport Agent (MTA): o próprio servidor smtp
- Mail Delivery Agent (MDA): responsável por passar o email a programas antivírus e antispam e sua posterior entrega na caixa postal do destinatário.

Instalando o sendmail

O sendmail foi o primeiro servidor de correio eletrônico criado e é, até o hoje, muito utilizado. Para a sua utilização no Debian vamos instalá-lo como segue.

```
# apt-get install sendmail sendmail-bin rmail sasl2-bin openssl
```

A configuração completa do sendmail é bem complexa porém você deve conhecer as configurações básicas para identificar domínios, gerenciar usuários e filas de email além de criar redirecionamentos.

Redirecionamentos de email - aliases

O primeiro passo é identificar os redirecionamentos básicos de um domínio, contas de email que devem existir e devem ser entregues a um usuário local. Estes emails são criados como entradas no arquivo `/etc/mail/aliases` e indicam a quem devem entregar as mensagens destinadas a estas contas. Os seguintes emails são definidos pela RFC2142 e devem existir em seu domínio. O ideal é que estes sejam criados como redirecionamentos e enviados a usuários reais de seu domínio. Veja um exemplo do arquivo `aliases` a seguir:

```
# Redirecionamentos criados pelo sendmail após a instalação
mailer-daemon: postmaster
postmaster: root
nobody: root
hostmaster: root
usenet: root
news: root
webmaster: root
www: root
ftp: root
abuse: root
noc: root
security: root

# mensagens para o root devem ser dirigidas a um usuário real
root: debian

# Redirecionamentos solicitados pela RFC 2142
info:          postmaster
support:       postmaster
abuse:         postmaster

# problemas com a infraestrutura de rede
noc:           root

# endereço para reportar problemas de segurança (deve ser em inglês)
security:      root

# administrador do servidor DNS
hostmaster:    root

# administrador http/web
www:           webmaster
webmaster:     root

# administrador FTP
ftp:           root

# grupos comumente utilizados
tech:          postmaster
```

Após a criação deste arquivo devemos gerar a base de trabalho (indexada) a partir dele com o comando:

```
# sendmail -bi
```

Permitindo o envio de mensagens pelo servidor

Por padrão apenas o localhost pode enviar mensagens pelo sendmail. Devemos alterar isto de forma que nossa rede possa fazer uso do servidor.

/etc/mail/access

```
# a melhor forma de identificar os clientes é através de domínios DNS
Localhost      RELAY
intranet       RELAY

# ou por endereçamento IP (rede local 192.168.2.0/24) – menos recomendável
192.168.2      RELAY
```

Agora geramos o arquivo indexado:

```
# makemap hash /etc/mail/access < /etc/mail/access
```

Configurando o domínio

Até o momento foram criadas configurações complementares, ainda não indicamos por qual domínio o sendmail irá responder e receber emails. Poderíamos alterar o arquivo de configuração /etc/mail/sendmail.cf, porém isto é totalmente não recomendado. A equipe do sendmail desenvolveu scripts que tornam esta tarefa mais simples e segura. As seguintes definições devem ser incluídas ou alteradas no arquivo /etc/mail/sendmail.mc.

```
# alterar linha 98
MASQUERADE_AS(`empresa.com.br')dnl
```

Para ativarmos esta configuração podemos rodar o comando sendmailconfig ou o comando m4 como no exemplo a seguir.

```
# sendmailconfig
ou
# m4 sendmail.mc > sendmail.cf
```

Iniciando o sendmail

A maioria das distribuições vem com scripts prontos para a inicialização do servidor sendmail, mas também é possível iniciá-lo pelo comando sendmail. O padrão é utilizar o init script:

```
# /etc/init.d/sendmail start
```

Mas o sendmail também pode ser iniciado com o comando sendmail e verificando o log:

Servidor web

O principal servidor web em uso na Internet é o Apache, atingindo em 60% e 70% de todos os servidores instalados em todas as plataformas (Linux, Unix, Windows e outros). O apache possui duas séries atualmente em desenvolvimento: a série 1.3 e a série 2.0. Com características e funcionalidades diferentes, as duas se mantém em uso porém a série anterior vem sendo gradualmente substituída pela nova.

Instalando o apache

No processo de instalação do apache devemos escolher a série a ser utilizada. Caso você esteja configurando um novo equipamento, opte pela série 2. Caso esteja apenas atualizando um servidor existente e não possa migrar todos os recursos instalados para a nova versão, opte pela série atualmente em uso.

Instalando o apache2 já com suporte a php4 e perl:

```
# apt-get install apache2-mpm-prefork apache2-utils libapache2-mod-php4 \
libapache2-mod-perl2
```

Logo após a instalação, o apache é iniciado automaticamente. Abra seu navegador e aponte para o seu ip para verificar se tudo correu bem.

Configuração básica do apache

Todos os arquivos que o apache utiliza estão localizados em /etc/apache2. Os seguintes arquivos e diretórios são utilizados:

ARQUIVO OU DIRETÓRIO	DESCRIÇÃO
apache2.conf	Configuração do servidor apache. Em outras distribuições fica em conf/httpd.conf
conf.d/	Diretório com configurações adicionais
mods-*/	Diretório com configuração e ativação de módulos
ports.conf	Portas nas quais o apache estará aceitando conexões
sites-*/	Diretório com configuração e ativação de sites

É comum que um único servidor Apache hospede vários sites web, desta forma, de acordo com a conexão solicitada pelo cliente seu servidor deverá assumir um nome diferente. Isto é controlado pelas configurações de cada site. O Apache no Debian já traz uma configuração inicial separada, que permite o uso de múltiplos domínios de forma mais simples. Estas diretivas são:

DIRETIVA	DESCRIÇÃO
ServerAdmin email@dominio	E-mail do administrador do site, normalmente é o email webmaster@domínio
ServerName site	Nome pelo qual o site será acessado, pode ser: www.dominio.com.br webmail.dominio.com.br etc...
DocumentRoot diretório	Diretório onde serão colocadas as páginas web servidas pelo site. Representa um caminho absoluto no servidor, por exemplo: /var/www/sites/dominio.com.br

O configuração padrão do servidor Apache no Debian pode ser consultada pelo arquivo /etc/apache2/sites-available/default.

Trecho do arquivo /etc/apache2/sites-available/default

```
...
ServerAdmin webmaster@localhost
DocumentRoot /var/www/
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
<Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
    RedirectMatch ^/$ /apache2-default/
</Directory>
...
```

A diretiva Options identifica quais ações poderão ser realizadas ao acessar a pasta especificada, algumas destas estão na tabela seguinte.

OPÇÃO	DESCRIÇÃO
Indexes	Lista o conteúdo da pasta se não houver um index.html
FollowSymLinks	Segue links simbólicos pelo sistema de arquivos
ExecCGI	Permite executar scripts CGI

Outro fator importante é poder controlar de onde será permitido o acesso às páginas do site através da diretiva `Order`. Seu funcionamento determina que ordem seguir, se permitir (`Allow`) ou negar (`Deny`) o acesso e a partir de onde. As possibilidades para localização são:

- `Allow from 127.0.0.1` – determinando um ip específico
- `Allow from .com.br` – permitindo acesso a partir de um domínio
- `Deny from 192.168.2.0/24` – negando uma rede
- `Deny from All` – negando tudo

Estas opções podem ser utilizadas tanto para `Allow` quanto para `Deny`.

Acesso autenticado

O controle de acesso pode ser feito diretamente no diretório a ser protegido ou através da configuração da diretiva `Directory`. A forma mais segura é através desta diretiva.

Trecho do arquivo `/etc/apache2/sites-available/default`

```
...
<Directory /var/www/restrito>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    AuthType Basic
    AuthName "Acesso controlado - Identifique-se"
    AuthUserFile /etc/apache2/usuarios
    Require valid-user
</Directory>
...
```

Demos criar o arquivo de usuários com as senhas. Isto é feito com o comando `htpasswd`.

```
# htpasswd -c /etc/apache2/usuarios debian
New password:
Re-type new password:
Adding password for user debian
```

Em seguida reinicie seu Apache e acesse a url `http://localhost/restrito`

Servidor de arquivos NFS

Compartilhamento de arquivos é um dos recursos básicos de qualquer rede. No Linux este serviço é provido pelo NFS – Network File System. O serviço NFS permite que estações rodando Linux ou sistemas POSIX acessem arquivos no servidor de forma transparente, mantendo as mesmas informações de proprietários e grupos como se estivessem acessando pelo disco local.

Acessando um servidor NFS

Estações acessam os dados compartilhados em um servidor NFS através do comando `mount`. Sua sintaxe e principais opções são:

mount [opções] servidor:compartilhamento diretório-local	
OPÇÕES	DESCRIÇÃO
<code>rw ro</code>	Acesso read-write (default) ou read-only
<code>hard</code>	Tenta acesso ao servidor até que este responda (padrão)
<code>soft</code>	Tentar montar o compartilhamento uma vez e caso haja falha permite o time-out da operação
<code>timeo</code>	Define o time-out da operação em minutos
<code>retry</code>	Número de tentativas de montagem
<code>intr</code>	Permite o cancelamento da operação
<code>bg</code>	Após uma falha na montagem tenta a operação em segundo plano

Um procedimento de montagem padrão deve ser como este:

```
# mount -t nfs -o bg,soft,timeo=1,intr instrutor:/exemplos /exemplos
```

Todos os mapeamentos existentes em um equipamento são controlados pelo arquivo `/etc/fstab`. Nele são listadas tanto as partições locais quanto os servidores remotos que devem estar disponíveis logo na inicialização do equipamento. Um mapeamento local deve ser como este:

```
# discos locais
/dev/hda1          /boot      ext3 defaults      0  2
/dev/hda5          swap       swap defaults     0  0
/dev/hda6          /          ext3 defaults     0  1

# servidores remotos
192.168.2.200:/exemplos /exemplos  nfs  bg,soft,timeo=1,intr  0  0
192.168.2.200:/home   /home     nfs  bg,soft,timeo=1,intr  0  0
```

Compartilhando dados com NFS

O processo de criação de compartilhamentos é gerado pelo servidor NFS que deve ser instalado em nosso equipamento.

```
# apt-get install nfs-user-server
```

O Compartilhamento de diretórios é feito pelo arquivo /etc/exports, como segue:

```
# Diretório    cliente(opções)
/home         192.168.2.0/24(rw,sync) NO_ROOT_SQUASH -> root LOCAL (DIREITO DE ACESSO LOCALMENTE)
/exemplos     *.sala2(ro,async)      NO_ROOT_SQUASH -> 
```

GRANA EM NOTES → ASYNCRONO

Identificar a rede que poderá montar o compartilhamento pode utilizar os seguintes critérios:

- ip ou hostname (192.168.2.200 ou instrutor.sala2): apenas este equipamento poderá montar o recurso
- domínio dns (*.sala2): todas as máquinas que estiverem cadastradas no DNS como parte deste domínio terão acesso
- endereço de rede/máscara (192.168.2.0/24 ou 192.168.2.0/255.255.255.0): todas as máquinas na faixa especificada terão acesso ao servidor

NFS – Referências

NFS HowTo – TLDP (<http://nfs.sourceforge.net/nfs-howto/>)

Servidor de arquivos SAMBA

O Samba é um serviço que implementa o protocolo SMB (Server Message Block) utilizado em compartilhamentos do tipo Windows®. Além do SMB o Samba também implementa o CIFS (Common Internet File System) utilizado em redes Microsoft® mais recentes.

O servidor Samba oferece os seguintes recursos:

- Atua como Domain Controller padrão NT4 para estações Windows®.
- Pode trabalhar como servidor primário ou secundário de outros servidores Samba
- Pode atuar como member server de uma rede Windows (padrão NT4 ou trabalhando como client ADS)
- Pode atuar como um simples compartilhamento padrão w9x.

Os serviços disponibilizados pelo Samba por servidor envolvem:

- smbd: Autenticação de usuários e compartilhamento de arquivos e impressoras
- nmbd: Resolução de nomes netbios como client ou atuando como servidor WINS

Instalando o Samba

Para a instalação completa do Samba necessitamos tanto da parte server como da parte client, fornecida pelos seguintes pacotes.

```
# apt-get install samba smbclient smbfs linpopup swat samba-doc
```

Durante a instalação no Debian, será perguntado qual o grupo de trabalho ao qual você estará conectado, se deseja utilizar senhas criptografadas e integrar o dhcp com o samba. Indique o grupo de trabalho e aceite o valor padrão das outras opções. Durante nossa configuração estaremos alterando estes dados.

Configuração básica do Samba

O Samba permite a integração com ambientes Windows de acordo com as configurações determinadas em seu arquivo `/etc/samba/smb.conf` que determina a forma como o servidor irá se comportar em sua rede.

Este arquivo de configuração é composto por três seções especiais, sendo apenas a primeira delas obrigatória.

- **global** – onde são cadastradas as informações de domínio, nível de segurança, tipo de senhas a utilizar entre outros
- **homes** – diretórios pessoais de cada usuário cadastrado no equipamento
- **printers** – compartilhamento automático de todas as impressoras cadastradas

A configuração do samba na seção `[global]` envolve definir as seguintes diretivas:

- **workgroup**: grupo de trabalho ou domínio do samba
- **server string**: comentário sobre o uso do equipamento
- **wins server**: quem é o servidor wins da rede
- **name resolve order**: ordem de pesquisa por nomes na rede
- **log file**: onde serão armazenados os logs do samba
- **max log size**: tamanho máximo dos logs
- **security**: nível de segurança para a identificação de usuários

O compartilhamento especial `[printers]` disponibiliza automaticamente todas as impressoras que estiverem cadastradas no print server local. Desta forma é necessário indicar qual é o servidor utilizado na seção `[global]`.

- **printing**: indica o servidor de impressão em uso
- **printcap name**: onde pode ser encontrada a lista de impressoras

Para os compartilhamentos `[homes]` e `[printers]` as diretivas abaixo devem ser definidas:

- **path**: caminha para a pasta do compartilhamento
- **browsable**: se o compartilhamento será visível pelo ambiente de rede
- **writable**: define um compartilhamento de disco
- **printable**: define um compartilhamento de impressora
- **create mask**: permissões de criação de arquivos
- **directory mask**: permissões de criação de diretórios
- **public**: se necessita de senha para acesso ao compartilhamento

Colocando todas as opções acima no arquivo `/etc/samba/smb.conf` temos:

```
[global]
workgroup = SALA2
server string = Servidor Samba
wins server = 192.168.198.1
name resolve order = lmhosts host wins bcast
log file = /var/log/samba/log.%m
max log size = 10
security = share
printing = cups
printcap name = cups

[homes]
comment = Home Directories
browseable = Yes
writable = no
create mask = 0700
directory mask = 0700
guest ok = Yes

[printers]
comment = All Printers
browseable = no
path = /tmp
printable = yes
public = no
writable = no
create mask = 0700
```

Para criar um compartilhamento adicional você deve incluir ao final deste arquivo uma nova seção com o nome com que deverá ser acessado e as opções a serem usadas, além de criar a pasta local a ser compartilhada, caso esta não exista. Vamos criar um compartilhamento chamado **dados** da pasta **/dados**. Inicialmente crie a pasta com permissões completas.

```
# mkdir /dados -m 0777
```

Agora acrescente ao arquivo **smb.conf** a descrição do compartilhamento abaixo.

```
[dados]
comment = Dados compartilhados
path = /dados
browseable = Yes
writable = Yes
public = Yes
```

Recarrega as configurações do samba e teste o acesso com o programa **smbclient**.

```
# /etc/init.d/samba reload
#
# smbclient //alunoN/dados -N
```

Caso você queira mapear uma pasta remota em um diretório local você pode utilizar o comando **smbmount** como abaixo.

```
# smbmount \\\\debian\\\\dados pasta-local/ -o guest
```

Servidor DNS

Nosso acesso a sites na Internet é feito utilizando nomes, porém nossos equipamentos necessitam de informações mais diretas para realizar estas conexões. Eles utilizam endereços IP que atuam como CEP's indicando a localização exata de cada servidor na web.

Converter nomes de sites como www.green.com.br para o seu endereço IP é a tarefa dos DNS. O servidor mais utilizado na internet é o BIND – Berkeley Internet Name Domain.

No servidor, cada domínio cadastrado é tratado como uma zona que armazena os nomes de servidores e qual o ip corresponde a cada um deles bem como dados especiais como a identificação do servidor DNS (NS) e do servidor de troca de emails (MX). São criadas ainda zonas reversas por onde é possível localizar um servidor baseado em seu endereço ip.

Tipos de configuração do BIND

O Bind pode assumir os seguintes tipos de configuração:

- Servidor principal (master)
Onde são cadastrados todos os dados do domínio, qual o DNS, o MX e o ip de cada um dos servidores disponíveis (www, mail, smtp, pop, webmail, etc...).
- Servidor secundário (slave)
Mantém uma cópia dos dados do servidor principal e responde pelo domínio junto com este, dividindo sua carga e garantindo a disponibilidade da informação.
- Servidor cache para navegação
Não responde por qualquer domínio porém é capaz de identificar os endereços ips solicitados a ele.

Arquivos de configuração do bind

Vários arquivos de configuração compõem o servidor bind. Todos estão localizados no diretório /etc/bind no Debian. Sua função e uso são:

ARQUIVO	Uso
named.conf	Configurações do servidor BIND define seu funcionamento básico e indica arquivos com dados comuns como localhost
named.conf.local	Dados criados pelo administrador para as zonas que serão cadastradas no servidor.
named.conf.options	Opções do servidor BIND, como diretório de trabalho e porta
db.*	Arquivos com os dados de cada zona cadastrada no servidor como os domínios externos, internos ou locais.

ARQUIVO	USO
hint zone	Arquivo contendo o endereço dos Root Servers utilizados para resolução de nomes na Internet. Seu arquivo pode se chamar: named.ca, named.root ou db.root

NOTA

Em distribuições como Mandriva, Fedora e SuSe, estes arquivos estão em /etc/named.conf e /var/named ou /var/named/chroot.

Configurando o BIND como um servidor cache com forward

O uso mais comum para um servidor DNS é a resolução de nomes para navegação na Internet ou para o envio de emails. Isto é feito criando uma zona chamada hint na configuração de seu servidor como abaixo:

```
zone "." {
    type hint;
    file "/etc/bind/db.root";
};
```

A existência desta zona em sua configuração fará com que, toda vez que este servidor receba uma solicitação para um domínio que ele não hospeda, seja disparada uma pesquisa aos Root Servers para localizar o dados solicitado. No Debian esta configuração já é padrão.

A funcionalidade de forward representa que toda solicitação que o servidor local não conheça deve ser encaminhada a um servidor remoto antes de efetuar uma pesquisa nos Root Servers. Isto é feito acrescentando-se, por exemplo, os servidores de seu provedor de acesso ao arquivo named.conf.options como abaixo.

Trecho de /etc/bind/named.conf.options

```
forwarders {
    dns1; dns2;
};

forward first;
```

Reinic peace seu servidor Bind, teste a resolução de nomes e veja o resultado.

```
# /etc/init.d/bind9 restart
#
# dig @localhost www.uol.com.br
```

Para que seja possível você utilizar seu servidor DNS de forma transparente, cadastre-o como servidor preferencial no arquivo `/etc/resolv.conf`.

```
# sufixo de pesquisa
search informe-o-domínio-local-aqui
# lista de servidores em ordem preferencial
nameserver 127.0.0.1
```

Com isto você será capaz de executar comandos como o `dig` sem a necessidade de informar o servidor a utilizar. Todo equipamento Linux já vem pré-configurado para pesquisar o nome de equipamentos na seguinte ordem: `/etc/hosts` e em seguida servidor dns.

Esta ordem é controlada pelo arquivo `/etc/nsswitch.conf`. Se houver a necessidade de forçar a pesquisa apenas pelo dns ou apenas pelo arquivo local `/etc/hosts` você pode alterar a diretiva `hosts` apresentada abaixo.

```
# /etc/nsswitch.conf

passwd:      compat
group:       compat
shadow:      compat

hosts:        files dns
networks:    files

protocols:   db files
services:    db files
ethers:      db files
rpc:         db files

netgroup:    nis
```

Em versões mais antigas do Linux esta ordem é controlada pelo arquivo `/etc/host.conf`, na diretiva `order` como segue:

```
order hosts,bind
multi on
```

Configurando o BIND como servidor autoritário de domínio

Dizer que seu servidor DNS será autoritário por um domínio indica que este servidor irá responder por este domínio e poderá ser cadastrado no registro.br. Este cadastramento envolve a criação de uma zona direta (resolvendo nomes para endereços ip) e uma zona reversa (resolvendo endereços ip para nomes). Nossa primeira etapa será a criação de uma zona interna para uso local pela empresa.

Trecho de /etc/bind/named.conf.local

```
...
zone "intranet" {
    type master;
    file "/etc/bind/db.intranet";
};

zone "2.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192.168.2";
};
```

Toda zona reversa tem, em seu nome, o endereço de rede ao contrário e o sufixo `in-addr.arpa`. Como nossa rede local é `192.168.2.0/24` utilizamos a parte fixa deste endereço (`192.168.2`), de traz para frente antes do sufixo padrão.

Agora vamos criar os arquivos com as informações de máquinas. Nosso exemplo levará em conta as seguintes máquinas e ip's:

- `dc.intranet` → `192.168.2.1`
- `ns.intranet` → `192.168.2.100`
- `mail.intranet` → `192.168.2.200`
- `pop.intranet` → `192.168.2.201`
- `smtp.intranet` → `192.168.2.202`
- `www.intranet` → `192.168.2.203`
- `gateway.intranet` → `192.168.2.254`

Criando o arquivo db.intranet de resolução direta:

```
$TTL 12H      ; tempo de validade dos registros
@   IN  SOA  ns.intranet. hostmaster.intranet. (
              2006080101 ; número de série
              15M       ; checar por alterações a cada
              5M        ; em caso de falha tentar novamente a cada
              2W        ; expira em
              12H       ; tempo de validade de um registro
)

; registros especiais
@           IN    MX    10 mail.intranet.
@           IN    NS    ns.intranet.

; nomes de máquinas
dc          IN    A     192.168.2.1
ns          IN    A     192.168.2.100
mail        IN    A     192.168.2.200
pop         IN    A     192.168.2.201
smtp        IN    A     192.168.2.202
www         IN    A     192.168.2.203
gateway     IN    A     192.168.2.254
```

Criando o arquivo db.192.168.2 de resolução reversa:

```
$TTL 12H      ; tempo de validade dos registros
@   IN  SOA  ns.intranet. hostmaster.intranet. (
              2006080101 ; número de série
              15M       ; checar por alterações a cada
              5M        ; em caso de falha tentar novamente a cada
              2W        ; expira em
              12H       ; tempo de validade de um registro
)

; registros especiais
@           IN    NS    ns.intranet.

; ips de máquinas
1            IN    PTR   dc.intranet.
100         IN    PTR   ns.intranet.
200         IN    PTR   mail.intranet.
201         IN    PTR   pop.intranet.
202         IN    PTR   smtp.intranet.
203         IN    PTR   www.intranet.
254         IN    PTR   gateway.intranet.
```

TESTAR OS ARQUIVOS

NAMED-checkzone : flavio.com.br /etc/bin/flavio.com.br

NAMED-checkzone : 2.168.192.in.ADDR.ARPA /etc/bin/192.168.2

Para podermos testar se nosso servidor está funcionando, vamos reiniciar o bind tentar um teste simples com o ping. Não podemos esquecer de alterar nosso arquivo /etc/resolv.conf que deverá apontar para o nosso próprio servidor.

```
# cat /etc/resolv.conf
search intranet
nameserver 127.0.0.1
#
# /etc/init.d/bind9 restart
Stopping domain name service: named.
Starting domain name service: named.
#
# ping -c2 dc.intranet
PING dc.intranet (192.168.2.1) 56(84) bytes of data.
64 bytes from dc.intranet (192.168.2.1): icmp_seq=1 ttl=64 time=1.04 ms
64 bytes from dc.intranet (192.168.2.1): icmp_seq=2 ttl=64 time=0.333 ms

--- dc.intranet ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.333/0.687/1.042/0.355 ms
```

Para que estes arquivos sejam adaptados para um domínio válido na Internet, basta que você substitua as ocorrências de intranet por dominio.com.br e altere o endereçamento interno de 192.168.2.X para os endereços válidos de seu domínio.

Criando um servidor DNS Slave

A função de um servidor SLAVE é manter uma cópia de todos os dados cadastrados no servidor principal, verificando periodicamente se ocorreram modificações da zone e atualizando-a localmente se necessário.

A configuração para este tipo de servidor envolve alterar tanto o servidor master quanto o servidor slave. No servidor principal iremos informar quais servidores podem solicitar uma cópia da zona cadastrada e no servidor secundário iremos informar para quais zonas atuamos como slave qual o servidor principal daquela zona.

Alterações no arquivo /etc/bind/named.conf.local do servidor master:

```
...
zone "intranet" {
    type master;
    file "/etc/bind/db.intranet";
    allow-transfer { 192.168.2.101; };
};

zone "2.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192.168.2";
    allow-transfer { 192.168.2.101; };
};
```

*TESTANDO
MANEJO DE ZONAS*

Criação da zona no servidor slave no arquivo /etc/bind/named.conf.local:

```
zone "intranet" {  
    type master;  
    file "/etc/bind/db.intranet";  
    masters { 192.168.2.100; };  
};  
  
zone "2.168.192.in-addr.arpa" {  
    type master;  
    file "/etc/bind/db.192.168.2";  
    masters { 192.168.2.100; };  
};
```

DNS – Referências

Internet Systems Consortium (<http://www.isc.org>)

djbdns: Domain Name System Tools (<http://cr.yp.to/djbdns.html>)

How does DNS work? (<http://cr.yp.to/djbdns/intro-dns.html>)

DHCP – Dynamic Hosts Configuration Protocol

Uma das tarefas mais árduas no gerenciamento de uma rede é a atribuição e manutenção do endereçamento dos hosts. Em rede de pequeno porte (até 20 máquinas) configurar os equipamentos com ip fixo é comum, porém pode causar pequenos contratemplos quanto for necessária a troca de gateway ou dns. Para automatizar esta tarefa foi criado o protocolo DHCP que atribui estes dados ao equipamento cliente, a partir de uma solicitação deste.

Instalando e configurando o DHCP Server

O servidor DHCP no Debian é fornecido pelo pacote `dhcp3-server` que deve ser instalado.

```
# apt-get install dhcp3-server
```

A configuração do DHCP Server envolve definições básicas válidas para o servidor e informações sobre a concessão de endereços IP para a rede local. As diretivas abaixo representam os parâmetros mais comumente utilizados.

OPÇÃO	DESCRIÇÃO
subnet	Define em qual rede o servidor irá atuar
range	Qual a faixa de endereçamento deverá ser concedida
domain-name-servers	Servidores de dns a informar aos clients
routers	Qual o gateway da rede
domain-name	Qual o sufixo dns de pesquisa de nomes
max-lease-time	Tempo máximo de uso das configurações recebidas
default-lease-time	Tempo padrão de uso das configurações recebidas
subnet-mask	Máscara de rede a ser utilizada
broadcast-address	IP de broadcast da rede
ntp-servers	Servidor de atualização de hora
time-offset	Diferença de fuso horário
netbios-name-servers	Servidor WINS da rede

Uma configuração padrão do servidor DHCP para a rede 192.168.2.0/24 fornecendo IP's na faixa de 101 a 150 é esta:

/etc/dhcp3/dhcpd.conf

```
# nome do servidor
server-identifier      dhcp.intranet;

# tipo de servidor
authoritative;

# tempo de concessão: 1 semana
default-lease-time     604800;

# tempo máximo de concessão: 2 semanas
max-lease-time         1209600;

# configurações específicas da faixa de concessão
subnet 192.168.0.0 netmask 255.255.255.0 {
    range               192.168.2.101 192.168.2.150;
    option domain-name   "intranet";
    option domain-name-servers 192.168.2.200;
    option netbios-name-servers 192.168.2.200;
    option routers        192.168.2.254;
    option subnet-mask     255.255.255.0;
    option broadcast-address 192.168.2.255;
}
```

Se for necessário que algum equipamento possua um ip fixo, você pode definir que o servidor irá conceder sempre o mesmo ip para um determinado equipamento, baseado em seu endereço físico (MAC Address). A configuração ser acrescentada ao arquivo acima é esta:

```
host instrutor {
    hardware 00:E0:29:42:7C:56;
    fixed-address 192.168.2.115;
}
```

INSTALAR O IPCA-16

~~CALCULAR~~

CALCULA O END. DE BROADCAST

Clientes dhcp

Há vários programas clientes para DHCP, com opções de uso e configuração específicas de cada um. Aqui veremos os principais clientes atualmente em uso. O cliente `dhclient` padrão no Debian sarge é o `dhclient versão 2`. Vamos abordar os seguintes clientes dhcp:

- dhcpcd
 - dhclient
 - pump

dhcpcd – dhcp client daemon

Para que este cliente possa ser utilizado é necessário instalar seu pacote

```
# apt-get install dhcpcd
```

Este programa solicita a configuração IP a ser utilizada pelo equipamento a um servidor local, armazenando as informações obtidas nos seguintes arquivos:

- /etc/dhcpc/resolv.conf resolução de nomes DNS
 - /var/lib/dhcpc/dhcpcd-<if>.info informações sobre a concessão ip
 - /var/lib/dhcpc/dhcpcd-<if>.cache cache de informações obtidas anteriormente

Seu funcionamento permite o uso de algumas opções pela linha de comando como na tabela abaixo.

dhcpcd [opções] interface	
OPÇÃO	Descrição
-d	Envia todas as solicitações para o SYSLOG
-k	Finaliza o dhcpcd desabilitando a interface
-D	Assume o domain name informado pelo servidor DHCP
-H	Utiliza o hostname informado pelo servidor
-t	Time out para a obtenção de ip

dhclient

mais novo

É o cliente padrão do servidor DHCP versão 3 e oferece maior estabilidade e facilidade de uso em relação a seus anteriores. Para que se possa utilizar esta versão, em substituição à versão 2 padrão do Debian é necessário instalá-lo como segue:

```
# apt-get install dhcpc3-client
```

Ele utiliza tanto o protocolo DHCP quanto BOOTP e é capaz de designar endereços estáticos caso a solicitação de ip falhe. Este cliente não necessita de parâmetros pela linha de comando pois ele obtém suas informações dos arquivos:

- /etc/dhcpc3/dhclient.conf configuração do client dhcp
- /var/run/dhclient.<if>.leases informações sobre a obtenção anterior de endereços utilizadas para renovação da concessão

pump

Este é o comando mais simples entre os clientes dhcp mas também aceita os protocolos DHCP e BOOTP. Para poder utilizá-lo, você deve instalá-lo manualmente.

```
# apt-get install pump
```

Parâmetros mais comuns passados pela linha de comando podem ser vistos na tabela abaixo.

pump [opções]	
OPÇÃO	DESCRIÇÃO
-i <iface>	Interface a ser ativada via dhcp
-s --status	Exibe o status da concessão dhcp
-d	Não altera o arquivo /etc/resolv.conf com os dados recebidos
-r	Libera a interface (release)
-R	Renova a concessão (renew)
-k	Finaliza o pump e libera a interface

DHCP – Referências

Internet Systems Consortium (<http://www.isc.org>)

Acesso remoto

Hoje em dia é comum falarmos em acessar um equipamento remotamente para manutenção, ainda mais com a popularização de programas como vnc® e radmin® para ambientes como estações Windows®. Contudo o acesso remoto em ambiente Linux não estão restritos à interface gráfica do usuário sendo freqüentemente realizado pela linha de comando para a manutenção de servidores. Entre os programas mais populares estão o telnet e o ssh.

O telnet foi uma das primeiras formas de acesso e oferece pouca segurança pois todo acesso realizado tem seus dados transmitidos em texto puro, ou seja, quando você informar seu usuário e senha de acesso, estes dados poderão ser capturados na rede por atacantes em potencial e facilmente explorados para conexões não autorizadas.

Para solucionar esta vulnerabilidade crítica do telnet foi desenvolvido o SSH, uma conexão remota criptografada que utiliza chaves assimétricas para iniciar a transferência de informações entre servidor e cliente.

SSH – Secure Shell

O SSH se tornou o software padrão para a conexão remota segura. Ele fornece um alto grau de criptografia além de autenticação segura. Sua implementação é extremamente simples e, por padrão, não permite o acesso de usuários privilegiados (root).

Entre suas características estão:

- Autenticação via chaves RSA ou DSA, SecurID, S/Key, Kerberos e TIS
- Permite o tráfego de aplicações gráficas em seu túnel tcp
- Pode ter seu tráfego compactado com gzip
- Permite o redirecionamento de portas do localhost pela conexão ssh estabelecida

Sua implementação é realizada instalando os pacotes a seguir.

```
# apt-get install ssh
```

Os arquivos de configuração do ssh envolvem a configuração do client e do server, sendo que normalmente somente o servidor é personalizado. As opções de funcionamento do client são passadas em linha de comando, permitindo ativar ou não uma opção quando desejado.

Fazendo uma conexão ssh pela primeira vez

Conectar a um equipamento remoto com ssh é simples:

```
debian@aluno1:~$ ssh 192.168.2.200
The authenticity of host '192.168.2.200 (192.168.2.200)' can't be established.
RSA key fingerprint is 8b:b6:1d:94:d6:ce:c9:c5:47:38:82:ec:47:90:14:76.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.2.200' (RSA) to the list of known hosts.
Password:
Last login: Sun Sep  2 18:13:30 2006 from 192.168.2.1
debian@instrutor:~$
```

Sempre que uma conexão ocorre pela primeira vez você será questionado a aceitar a chave de identificação do host remoto. Após aceitar este ID, será apresentado o prompt de senha para autenticar o usuário.

Por padrão, o ssh solicita a senha de um usuário com o mesmo login do usuário local. Se estou conectado como usuário debian na máquina local, será solicitada a senha do usuário debian na máquina remota. Para efetuar login remoto como outro usuário rode o comando com a opção **-l** ou informe o nome do usuário remoto antes do ip de conexão. Os dois exemplos a seguir têm a mesma funcionalidade.

```
debian@aluno1:~$ ssh -l marcos 192.168.2.200
debian@aluno1:~$ 
debian@aluno1:~$ ssh marcos@192.168.2.200
```

As principais opções do comando ssh estão listadas na tabela a seguir.

ssh [opções] [usuario@]host	
OPÇÃO	DESCRIÇÃO
-l	Especifica um login remoto
-C	Compacta o tráfego do comando ssh
-X	Habilita o encaminhamento do tráfego X11
-p PORTA	Permite a conexão na porta especificada
-1 -2	Força o uso da versão do protocolo ssh especificada
-q	Modo silencioso, apenas erros fatais são exibidos
-v	Modo detalhado

Copiando arquivos entre máquinas

O SSH permite, além da conexão remota a cópia de arquivos entre equipamentos com o comando `scp`. Seu funcionamento é similar ao comando `cp` do Linux, incluindo a informação de usuário e host de origem ou destino da cópia. Veja alguns exemplos:

Copiando dados da máquina local para um host remoto no diretório `/tmp`:

```
# scp /etc/adduser.conf 192.168.2.200:/tmp
```

Copiando um diretório remoto para a máquina local no diretório corrente:

```
# scp -r 192.168.2.200:/etc .
```

As principais opções do comando `scp` são:

scp [opções] [usuario@]host	
OPÇÃO	DESCRIÇÃO
-C	Compacta o tráfego do comando <code>scp</code>
-l USUARIO	Utiliza o usuário especificado para conexão
-P PORTA	Usa a porta especificada para a conexão

Usando chaves RSA/DSA com ssh

A suíte de aplicativos SSH permite o uso de chaves de criptografia assimétricas para a autenticação de usuários. As chaves do tipo RSA são utilizadas como autenticação de conexões em versão 1 do protocolo SSH e as DSA para a versão 2. Para gerar as chaves de autenticação utilizamos o comando `ssh-keygen` como segue.

```
# ssh-keygen -t das -b 1024
Generating public/private dsa key pair.
Enter file in which to save the key (/root/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_dsa.
Your public key has been saved in /root/.ssh/id_dsa.pub.
The key fingerprint is:
e2:1a:25:68:c5:8e:11:05:4a:c3:c2:62:e6:26:72:e0 root@debian
```

Serão criados dois arquivos com sua chave:

- `~/.ssh/id_dsa`: chave pessoal
- `~/.ssh/id_dsa.pub`: chave pública a ser cadastrada nos equipamentos remotos

No momento em que for solicitada a passphrase (senha), se você deixar isto em branco, sua conexão SSH da máquina para os equipamentos onde você cadastre esta chave não necessitarão de senha. Agora devemos colocar o conteúdo do arquivo `~/.ssh/id_dsa.pub`

dentro do arquivo `~/.ssh/authorized_keys` no equipamento onde queremos nos conectar.

Podemos copiar o arquivo para o host remoto, em seguida nos conectar a ele e depois colocar a chave dentro do arquivo `~/.ssh/authorized_keys`. O comando `ssh-copy-id` pode fazer tudo isto por nós.

```
$ cd ~  
$  
$ ls -a .ssh  
id_dsa  id_dsa.pub  known_hosts  
$  
$ ssh-copy-id -i .ssh/id_dsa.pub debian@192.168.2.200  
15  
Password:  
Now try logging into the machine, with "ssh 'debian@192.168.2.200'", and check  
in:  
  .ssh/authorized_keys  
to make sure we haven't added extra keys that you weren't expecting.  
$ ssh debian@192.168.2.200
```

SSH – Referências

Cientes para equipamentos Windows®:

PutTY (<http://www.chiark.greenend.org.uk/~sgtatham/putty/>)

TeraTerm (<http://hp.vector.co.jp/authors/VA002416/teraterm.html>)

TTSSH (<http://www.zip.com.au/~roca/ttssh.html>)

WinSCP (<http://winscp.net>)

OpenSSH (<http://www.openssh.org>)

Segurança básica

Tendo em mente que a segurança de um sistema é responsabilidade de todos os usuários, você como Administrador deve definir as políticas a serem adotadas e garantir a sua aplicação. Devem ser desenvolvidas políticas e ações que cubram todas as áreas dos sistemas de informação adotados pela empresa, envolvendo:

- análise de riscos
- medidas de segurança física
- auditoria
- respostas ativas

O que é segurança

Segurança é um assunto extenso que envolve o desenvolvimento, implementação e uso de medidas que demandam muito tempo e energia. Os pontos básicos na segurança de sistemas de informação envolvem:

- Eliminar todas as aplicações desnecessárias, removendo-as ou não instalando estes aplicativos
- Restringir o acesso aos recursos da rede
- Limitar o acesso a ferramentas administrativas apenas aos administradores

Softwares são conhecidos por possuírem falhas de segurança que são divulgadas a todo momento. Os principais sites sobre alertas de segurança são:

- Security Focus – <http://www.securityfocus.com>
- Cert Coordination Center – <http://www.cert.org>

Segurança Física

Toda política de segurança deve começar com o controle do acesso físico aos equipamentos de rede, em especial a servidores de missão crítica e de autenticação da rede. Determinar a localização física destes equipamentos influencia diretamente no grau de segurança inicial oferecido por uma rede para a empresa.

Se há acesso físico aos equipamentos de rede (servidores e ativos) não há segurança.

Segurança de software

As medidas iniciais para o controle de seu sistema operacional são variadas, sendo que as mais recomendadas no caso do sistema operacional Linux são:

- desabilite ou remova contas de usuários inativas
- sempre utilize senhas shadow em seus servidores
- verifique as permissões de arquivos importantes em seus servidores, em especial os localizados no /etc e cheque o uso de arquivos com permissões superiores a 666 em todos os sistemas de arquivos do servidor
- habilite firewalls locais que permitam apenas o tráfego a serviços pré-determinados, reduzindo o risco de conexões não autorizadas
- force o uso de servidores proxy para o acesso à Internet para um melhor controle das informações passadas e recebidas da Internet
- amplie os recursos de segurança local habilitando senhas fortes para usuários pelo PAM além de controlar o acesso aos recursos de hardware permitidos

Ferramentas de segurança

Verificar a integridade do sistema, e garantir a aplicação das políticas de segurança são tarefas árduas e maçantes que demandam muito tempo. Para facilitar estas tarefas utilize softwares especializados como:

- Nessus: análise remota de segurança de redes
- AIDE – Advanced Intrusion Detection Environment: permite a verificação da integridade do sistema
- SNORT: um sistema de detecção de tentativas de invasão à rede (NIDS)
- TAMU: uma série de scripts utilizados para ampliar a segurança de hosts individuais
- COPS: uma série de aplicativos que documentam o estado do servidor indicando ações corretivas que devem ser adotadas
- Bastille Linux: programa que amplia a segurança dos equipamentos Linux reconfigurando serviços e ativando recursos de firewall entre outros
- SELinux: uma série de aplicativos que implementam acl's mandatórias no kernel do Linux seguindo o modelo Linux Security Modules (LSM)

Password Shadow suite – Senhas

Até alguns anos atrás as senhas de usuários eram cadastradas no arquivo `/etc/passwd` junto com os dados de cadastramento das contas o que levava a ataques de obtenção de senhas por `ftp` e posteriormente ataques de força bruta sobre estes dados.

O uso de shadow passwords separou as senhas, tanto de usuários quanto de grupos, em arquivos específicos com permissões de acesso mais restritivas. As aplicações utilizadas para isto são:

- `pwconv`: separa as senhas de usuários do arquivo `/etc/passwd` para o `/etc/shadow`.
- `pwunconv`: retorna ao formato original
- `grpconv`: separa as senhas de grupo criando o arquivo `/etc/gshadow`
- `grpunconv`: retorna o arquivo `/etc/group` ao formato original

tcp wrapper

Implementa um nível extra de segurança na camada de rede verificando a origem dos acessos aos serviços locais, permitindo ou negando estes acessos. Implementa a checagem com base nos serviços a serem acessados, de acordo com os arquivos `/etc/hosts.allow` e `/etc/hosts.deny`. O formato destes arquivos é:

`<serviço>: <origem da conexão> <comandos>`

Exemplos de uso para o arquivo `hosts.allow`:

```
ALL: 127.0.0.1: ALL
imapd, ipop3d: 192.168.2.: ALL
ALL EXCEPT in.telnetd: ALL
```

Exemplos de uso do arquivo `hosts.deny`:

```
in.telnetd: ALL
ipop3d: ALL
```

Firewall básico

Na forma mais simples um Firewall pode ser definido como o controle de três tipos de tráfego de rede:

- Tráfego de entrada (INPUT): toda conexão que se destina ao equipamento
- Tráfego de saída (OUTPUT): toda conexão originada no equipamento
- Tráfego de passagem (FORWARD): toda conexão que passa pelo equipamento

Para cada um destes tipos de tráfego há três ações básicas a serem tomadas:

- ACCEPT: Aceitar a conexão e permitir o processamento do pacote de comunicação
- DROP: Descartar o pacote de comunicação
- REJECT: Recusar o pacote de comunicação

Além destes aspectos deve-se considerar, ainda, que todo Firewall baseado em iptables tem três tabelas básicas:

- FILTER: onde os pacotes de comunicação são aceitos ou rejeitados
- MANGLE: onde pode se alterar características do pacote como QOS entre outros
- NAT: permite o mascaramento de pacotes

Usando o iptables

Todo firewall é criado a partir de um script básico onde são cadastradas as regras a serem utilizadas pela rede. Por padrão todo firewall deve começar definindo uma política padrão a ser adotada pelo equipamento. A regra recomendada é negar tudo e permitir apenas o estritamente necessário.

Vamos definir algumas regras que permitirão que uma pequena rede acesse a Internet de forma irrestrita porém segura não permitindo nenhum tráfego de entrada vindo da Internet. Considere que a rede local é 192.168.2.0/24, que a rede local está conectada à placa eth0 e a conexão Internet está conectada à placa eth1. Crie um arquivo com as seguintes regras que poderão ser executadas a qualquer momento.

```
# Definindo a política padrão (-P)
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP

# Permitindo que a rede local acesse serviços no servidor local
iptables -A INPUT -s 192.168.2.0/24 -i eth0 -j ACCEPT

# Permitindo a saída de pacotes da rede local para a Internet
iptables -A FORWARD -s 192.168.2.0/24 -d 0.0.0.0/0 -j ACCEPT

# Habilitando o mascaramento da rede local na saída para a Internet
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -d 0.0.0.0/0 -j MASQUERADE -o
eth1
```

Neste exemplo as linhas 2, 3 e 4 definem que a política padrão é tudo fechado. A linha 7 permite que a rede local acesse qualquer serviço disponível no equipamento, sem restrição. A linha 10 permite que qualquer tráfego vindo da rede local com destino à Internet seja permitido. A última linha diz que todo tráfego vindo da rede local e que se destine à Internet utilize o ip do Firewall para navegar (mascare o ip).

ipfwadm e ipchains

Estes programas foram as ferramentas de Firewall utilizadas nas séries 2.0 e 2.2 do kernel do Linux.

Após o lançamento de uma nova série de kernel, a versão anterior tem seu suporte finalizado após 3 anos. O suporte oficial ao ipchains no kernel série 2.4 finalizou em Março/2004 não sendo mais atualizado desde então. Na série 2.6 existe apenas o suporte básico.

O ipfwadm da série 2.0 não teve suporte implementado nas séries 2.4 ou 2.6 do kernel do Linux.

Firewall – Referências

NetFilter – <http://www.netfilter.org>

Linux Advanced Routing and TC – <http://www.lartc.org>

The Linux Kernel – <http://www.kernel.org>

Segurança local – desativação de serviços

O primeiro passo para ampliar a segurança local é identificar os serviços em execução e desativar os que não são necessários. Podemos listar todos os serviços ativos no boot do equipamento, através dos links simbólicos dos diretórios /etc/rc?.d como a seguir.

```
# cd /etc
# ls rc2.d/S*
rc2.d/S10sysklogd      rc2.d/S20dirmngr    rc2.d/S21fam
rc2.d/S11klogd         rc2.d/S20exim4     rc2.d/S21nfs-common
rc2.d/S12alsa          rc2.d/S20inetd    rc2.d/S21sendmail
rc2.d/S14ppp           rc2.d/S20lpd      rc2.d/S25nfs-user-server
rc2.d/S15bind9         rc2.d/S20makedev   rc2.d/S89atd
rc2.d/S18portmap       rc2.d/S20samba    rc2.d/S89cron
rc2.d/S20cupsys        rc2.d/S20saslauthd  rc2.d/S91apache2
rc2.d/S20dbus-1        rc2.d/S20ssh      rc2.d/S99rmnologin
rc2.d/S20dhcp3-server  rc2.d/S20xinetd   rc2.d/S99stop-bootlogd
```

Identifique os serviços instalados e desabilite os que não são mais necessários. Um exemplo é o serviço lpd (servidor de impressão) acima, uma vez que temos também o servidor cups instalado e ativo. Para removê-lo da inicialização automática utilize o comando:

```
# update-rc.d -f lpd remove
update-rc.d: /etc/init.d/lpd exists during rc.d purge (continuing)
Removing any system startup links for /etc/init.d/lpd ...
/etc/rc0.d/K20lpd
/etc/rc1.d/K20lpd
/etc/rc2.d/S20lpd
/etc/rc3.d/S20lpd
/etc/rc4.d/S20lpd
/etc/rc5.d/S20lpd
/etc/rc6.d/K20lpd
```

Realize este mesmo procedimento com todos os serviços desnecessários ou remová-os com o comando apt-get remove.

Monitorando os logs do sistema

Acompanhar um log do sistema pode se tornar uma tarefa difícil em equipamentos em produção devido à quantidade de informações geradas nos logs.

Sempre que necessário verificar um destes arquivos do sistema de log, utilize um paginador como o comando less que permita efetuar pesquisas e rápida navegação dentro do arquivo. Se necessário um acompanhamento em tempo real do log, o comando abaixo permite isto.

```
# tail -f arquivo.log
```

Este comando permitirá que toda nova ocorrência seja enviada à tela simultaneamente à sua gravação em disco. Para um acompanhamento mais eficaz aplique um filtro procurando por trechos sugestivos como um endereço ip, um nome de usuário ou uma negativa de acesso.

Apêndice A – Configurações adicionais do Apache

No arquivo `/etc/apache2/apache2.conf` as seguintes diretivas podem ser utilizadas para controlar a carga e o desempenho do servidor na seção MPM prefork:

OPÇÃO	DESCRIÇÃO
<code>StartServers 5</code>	Número inicial de conexões permitidas ao servidor
<code>MinSpareServer 5</code> <code>MaxSpareServers 10</code>	Número mínimo de conexões disponíveis e número de máximo de conexões disponíveis em espera (sem uso)
<code>MaxClients 20</code>	Número máximo de conexões simultâneas aceitas pelo servidor. Se necessário aumentar o número de conexões simultâneas, instale o pacote <code>apache2-mpm-worker</code> como servidor básico

Um recurso adicional que nos dará informações sobre o status do servidor está, normalmente, desabilitado. Vamos habilitá-lo descomentando as linhas abaixo e alterando os locais de onde podemos acessar a página.

Trecho do arquivo `/etc/apache2/apache2.conf`

```
...
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
    Allow from 192.168.2.0/24
</Location>
...
```

Em seguida reiniciamos o apache e você poderá abrir seu navegador e apontar para `http://127.0.0.1/server-status`.

```
# apache2ctl restart
```



www.green.com.br