

¡Completamente revisado y actualizado!



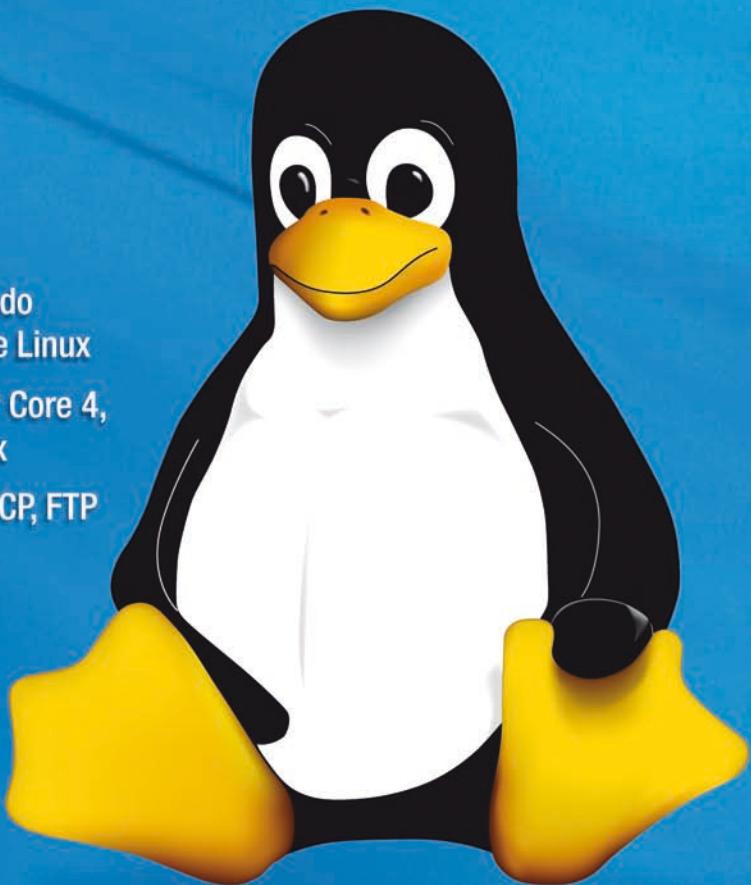
MANUAL DE ADMINISTRACIÓN DE LINUX

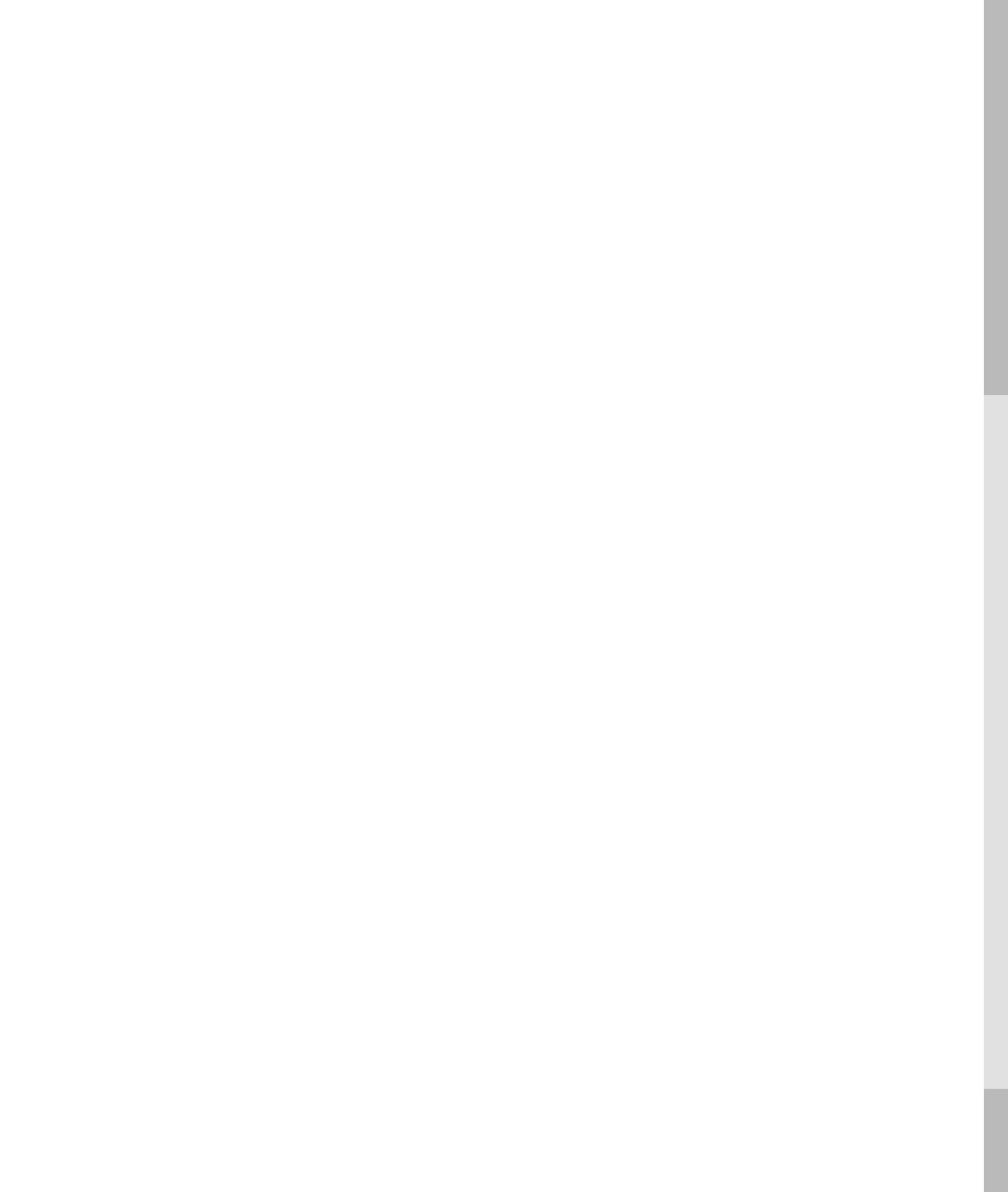
Steve Shah

Wale Soyinka

Cuarta edición

- Proporcione seguridad al sistema usando los firewalls y los filtros de paquetes de Linux
- Configure, ajuste y administre Fedora™ Core 4, SuSE Linux y Red Hat® Enterprise Linux
- Administre los servicios Web, DNS, DHCP, FTP y correo electrónico





Manual de administración de Linux

STEVE **SHAH** Y WALE **SOYINKA**

Traducción:

José Hernán Pérez Castellanos
Traductor profesional

Manuel F. Mejías Butrón

*Director de Ingeniería,
Investigación & Desarrollo Tecnológico Kreissontech*



MÉXICO • BOGOTÁ • BUENOS AIRES • CARACAS • GUATEMALA
LISBOA • MADRID • NUEVA YORK • SAN JUAN • SANTIAGO
AUCKLAND • LONDRES • MILÁN • MONTREAL • NUEVA DELHI
SAN FRANCISCO • SINGAPUR • ST. LOUIS • SIDNEY • TORONTO

Gerente de división: Fernando Castellanos Rodríguez
Editora de desarrollo: Cristina Tapia Montes de Oca
Supervisor de producción: Jacqueline Brieño Álvarez

Manual de administración de Linux

Prohibida la reproducción total o parcial de esta obra,
por cualquier medio, sin la autorización escrita del editor.



DERECHOS RESERVADOS © 2007 respecto a la primera edición en español por
McGRAW-HILL/INTERAMERICANA EDITORES, S.A. DE C.V.

A Subsidiary of The McGraw-Hill Companies, Inc.

Prolongación Paseo de la Reforma 1015, Torre A

Piso 17, Colonia Desarrollo Santa Fe,

Delegación Álvaro Obregón

C.P. 01376, México, D. F.

Miembro de la Cámara Nacional de la Industria Editorial Mexicana, Reg. Núm. 736

ISBN: 970-10-5882-8

Translated from the fourth English edition of
LINUX ADMINISTRATION A BEGINNER'S GUIDE

By: Steve Shah, Wale Soyinka

Copyright © MMV by The McGraw-Hill Companies, all rights reserved.

0-07-226259-1

1234567890

0987543216

Impreso en México

Printed in Mexico

Dedicado a mi familia, por su apoyo, paciencia y amor infinitos.

—Steve Shah

**Me agradaría dedicar las partes de este libro de las cuales soy responsable
a todos aquellos a quienes les gusta el software de fuente abierta
y a todos los que han contribuido a este tipo de software en una u otra forma.**

—Wale Soyinka

ACERCA DE LOS AUTORES

Steve Shah es el Director of Product Management en la división NetScaler de Citrix Systems, en donde es el responsable de las capacidades de conmutación Security y Layer 4-7 de los sistemas de entrega de aplicaciones, ganadoras de un premio de la compañía. Antes de su ingreso a NetScaler, Steve desempeñó un papel semejante en Array Networks, en donde hizo de todo, desde hacker en TCP/IP al nivel de núcleo hasta administración del producto. Antes de ingresar a Array Networks, fue miembro de Alteon Web Systems, en donde encabezó el esfuerzo de desarrollo de su producto acelerador SSL.

Además del libro *Manual de administración de Linux*, Steve ha sido colaborador para varias ediciones de *UNIX Unleashed*, *RedHat Linux Unleashed*, *Using Linux* y *Content Delivery Networks*. Obtuvo el grado de B. S. en Ciencia de la computación llevando como curso optativo Escritura creativa, y su M. S. en Ciencia de la computación, de la UC Riverside. Steve se ha dedicado a programar desde 1986; a la administración de sistemas desde 1992, y también a la administración del producto desde 2001.

Cuando Steve no está trabajando o desempeñándose como hacker en alguna parte de Linux, lo que le fascina (lo cual su media naranja, Heidi, argumentará que no hay tal), prueba sus habilidades en ser un mejor DJ y fotógrafo. El lector puede ver la página (*blog*) de Steve en <http://www.planetoid.org/blog>.

Wale Soyinka es consultor de Systems & Network Engineering, con varios años de experiencia en el campo. Obtuvo el grado de B. S. en Matemáticas/Estadística. Ha escrito materiales de capacitación en administración de Linux que se usa como parte del currículo en algunos Community Colleges en el Área de la Bahía. También es el autor de un manual para el laboratorio de proyectos, *Microsoft Windows 2000 Managing Network Environments*, el cual es parte de la serie de certificación de Microsoft publicada por Prentice Hall. En la actualidad participa en varias discusiones y diversos proyectos de fuente abierta. Sus intereses son muy amplios y variados.

CONTENIDO

Reconocimientos	xix
Introducción	xxi

Parte I

Instalación de Linux como servidor

▼ 1 Resumen técnico de las distribuciones de Linux y de Windows 2003	3
Aprendizaje acerca del sistema operativo Linux	4
¿Qué hay en torno del software libre y de GNU?	5
¿Qué es la GNU Public License?	6
Ventajas del software libre	7
Comprensión de las diferencias entre Windows y Linux	8
Usuarios únicos en comparación con usuarios múltiples y en comparación con usuarios de redes	8
El núcleo monolítico y el micronúcleo	9
Separación de la GUI y el núcleo	9
El Network Neighborhood	10
El Registry en comparación con los archivos de texto	11
Dominios y el Active Directory	12
Otras referencias	13

▼ 2 Instalación de Linux en configuración de servidor	15
Hardware y consideraciones ambientales	16
Diseño del servidor	16
Tiempo útil	18
Aspectos de la inicialización dual del sistema	18
Métodos de instalación	19
Instalación de Fedora Core Linux	20
Requisitos previos del proyecto	20
Realización de la instalación	21
Configuración inicial del sistema	40
Resumen	44
▼ 3 Instalación de software	45
El RED HAT Package Manager	46
Consulta con RPM (empezar a conocerse entre sí)	49
Instalación con RPM (frecuentarse)	52
Desinstalación con RPM (finalización de la relación)	55
Otras cosas que puede hacer con RPM	56
Administradores de paquetes GUI RPM	57
Compile e instale el software GNU	60
Obtención y desempaque del paquete	61
Búsqueda de la documentación (empezar a conocerse entre sí, una vez más)	62
Configuración del paquete	63
Compilación del paquete	64
Instalación del paquete	65
Prueba del software	65
Limpieza	66
Problemas comunes en la estructuración a partir de la fuente	66
Problemas con las bibliotecas	66
Cuando no hay script para configurar	67
Código fuente interrumpido	67
Resumen	67

Parte II**Administración con un solo anfitrión**

▼ 4 Administración de los usuarios	71
¿Qué constituye exactamente un usuario?	72
Dónde se guarda la información del usuario	72
El archivo /etc/passwd	73
El archivo /etc/shadow	77
El archivo /etc/group	78

Herramientas para administración de los usuarios	79
Administración de los usuarios con la línea de comandos	79
Administradores GUI de usuarios	83
Usuarios y los permisos de acceso	85
Comprensión de los programas SetUID y SetGID	86
Pluggable Authentication Modules (PAM)	86
Cómo funciona PAM	87
Archivos de PAM y sus ubicaciones	87
Configuración de PAM	88
Depuración de PAM	93
Un gran recorrido	93
Creación de usuarios con useradd	93
Creación de grupos con groupadd	95
Modificación de los atributos de los usuarios con usermod	95
Modificación de los atributos de los grupos con groupmod	96
Eliminación de grupos y usuarios con groupdel y userdel ..	96
Resumen	97
▼ 5 La línea de comandos	99
Una introducción a BASH	100
Control de tareas	101
Variables de entorno	102
Atajos de la línea de comandos	105
Expansión de los nombres de archivo	105
Herramientas para la documentación	107
El comando man	107
El sistema texinfo	109
Comprendiendo la formación de listas de archivos, de las propiedades y de los permisos	109
Formar listas de archivos: ls	109
Tipos de archivos y de directorios	110
Dispositivos de bloques	111
Dispositivos de caracteres	112
Tuberías nombradas	112
Cambiar la propiedad: chown	112
Cambiar el grupo: chgrp	113
Cambiar el modo: chmod	113
Administración y manipulación de archivos	116
Copiar archivos: cp	116
Mover archivos: mv	116
Vincular archivos: ln	117
Encontrar un archivo: find	117
Compresión de archivo: gzip	118
bzip2	118

Crear un directorio: mkdir	119
Eliminar un directorio: rmdir	119
Mostrar el directorio actual de trabajo: pwd	119
Archivo de cinta: tar	119
Concatenar archivos: cat	122
Presentar un archivo una pantalla a la vez: more	122
Utilización del disco: du	122
Mostrar la ubicación del directorio de un archivo: which	123
Localizar un comando: whereis	123
Espacio libre del disco: df	124
Sincronizar discos: sync	124
Movimiento de un usuario y su directorio inicial	125
Hacer una lista de procesos: ps	127
Mostrar una lista interactiva de procesos: top	129
Enviar una señal a un proceso: kill	130
Herramientas diversas	132
Mostrar el nombre del sistema: uname	132
Quién ha entrado: who	133
Variación de who: w	133
Comutar el usuario: su	133
Editores	134
vi	134
emacs	134
joe	135
pico	135
Normas	135
Resumen	136
 ▼ 6 Inicialización y apagado	137
Cargadores de inicialización	138
GRUB	138
LILO	148
Arranque	148
El proceso init	149
Scripts rc	150
Escritura de su propio script rc	151
Activación y desactivación de servicios	155
Desactivación de un servicio	157
Singularidades y fines de la inicialización y el apagado	158
fsck!	158
Inicialización en el modo de un solo usuario ("Recovery")	159
Resumen	160

▼ 7 Sistemas de archivos	161
La composición de los sistemas de archivos	162
Nodos i	162
Superbloques	163
ext3 y ReiserFS	164
¿Cuál sistema de archivos usar?	165
Administración de los sistemas de archivos	165
Montaje y desmontaje de los discos locales	165
Uso de fsck	171
Adición de un disco nuevo	173
Panorama general de las particiones	173
Convenciones acerca del nombramiento de los discos y particiones	174
Administración de volúmenes	174
Creación de particiones y volúmenes lógicos	175
Creación de los sistemas de archivos	184
Resumen	186
▼ 8 Servicios centrales del sistema	187
El servicio init	188
El archivo /etc/inittab	188
El comando telinit	191
xinetd e inetd	191
El archivo /etc/xinetd.conf	193
El demonio syslogd	199
Invocación de syslogd	199
El archivo /etc/syslog.conf	199
CRON	204
El archivo crontab	205
Edición del archivo crontab	206
Resumen	207
▼ 9 Compilación del núcleo de Linux	209
¿Qué es exactamente un núcleo?	210
Manera de hallar el código fuente del núcleo	211
Obtención de la versión correcta del núcleo	212
Desempaque el código fuente del núcleo	213
Estructuración del núcleo	213
Preparación para configurar el núcleo	215
Configuración del núcleo	216
Compilación del núcleo	219
Instalación del núcleo	220
Inicialización del núcleo	222
El autor mintió: ¡no funcionó!	223

Parchado del núcleo	224
Descarga y aplicación de parches	224
Resumen	226
▼ 10 Perillas y carátulas: el sistema de archivos proc	227
¿Qué se encuentra dentro del directorio /proc?	228
Pequeños ajustes a los archivos dentro de /proc	229
Algunas entradas útiles de /proc	229
Enumeración de entradas de /proc	231
Informes y ajustes comunes que se hacen con proc	232
Protección contra la inundación SYN	233
Aspectos acerca de servidores de alto volumen	234
Depuración de conflictos del hardware	235
SysFS	235
Resumen	237

Parte III**Seguridad y operación en red**

▼ 11 TCP/IP para administradores de sistemas	241
Las capas	242
Paquetes	242
TCP/IP y el modelo OSI	245
Encabezados	249
Ethernet	249
IP	251
TCP	254
UDP	257
Una conexión TCP completa	258
Apertura de una conexión	258
Transferencia de datos	259
Cierre de la conexión	260
Cómo funciona ARP	261
El encabezado ARP: ¡ARP también funciona con otros protocolos!	262
Unión de redes IP	263
Anfitriones y redes	263
Netmasks	265
Encaminamiento estático	266
Encaminamiento dinámico con RIP	268
Escudriño de tcpdump	274
Unas cuantas notas generales	274
Uso de tcpdump para observar la ruta de un rastro	276

¿Por qué es lento el DNS?	277
Trazo de gráficas de singularidades y fines	279
Resumen	282
▼ 12 Configuración de la red	285
Módulos e interfaces de red	286
Uso de ifconfig para configurar direcciones IP asignadas	287
Uso simple	288
Estructuración de las NIC en el momento de la inicialización	289
Parámetros adicionales	291
Uso de las rutas	292
Uso simple	292
Presentación de las rutas	293
Router Linux sencillo	295
Encaminamiento con rutas estáticas	295
Cómo Linux elige una dirección IP	298
Resumen	299
▼ 13 Configuración del firewall de Linux	301
Cómo funciona el Netfilter	302
Un compendio de NAT	303
Protocolos amigables para NAT	306
Cadenas	306
Instalación de Netfilter	309
Activación de Netfilter en el núcleo	310
Compilación de las tablas de IP	312
Configuración de Netfilter	313
Guardado de su configuración de Netfilter	314
El comando iptables	315
Soluciones del libro de recetas	322
NAT de tres líneas de Rusty	322
Configuración de un firewall simple	323
Resumen	325
▼ 14 Seguridad local	327
Fuentes comunes de riesgo	329
Programas SetUID	329
Procesos innecesarios	331
Programas que se ejecutan como raíz	333
Acceso otorgado a los usuarios	333
Mitigación del riesgo	336
Uso de chroot	336
SELinux	339

Monitoreo de su sistema	339
Registro cronológico	339
Uso de ps y netstat	340
Uso de df	340
Listas de correo	340
Resumen	341
▼ 15 Seguridad en la red	343
TCP/IP y seguridad en la red	344
Importancia de los números de los puertos	344
Seguimiento del rastro de los servicios	345
Uso del comando netstat	345
Implicaciones de seguridad de la salida de netstat	346
Interrupción de servicios	348
Monitoreo de su sistema	350
Forma de hacer el mejor uso de syslog	350
Monitoreo del ancho de banda con MRTG	351
Manejo de los ataques	351
Herramientas para la seguridad en la red	352
Resumen	354

Parte IV**Servicios de Internet**

▼ 16 DNS	357
El archivo de anfitriones	358
Primero, entender cómo trabaja DNS	359
Dominios y convencionalismos sobre nombramiento de anfitriones	359
Subdominios	361
El dominio in-addr.arpa	363
Tipos de servidores	363
Instalación de un servidor DNS	365
Comprensión del archivo de configuración BIND	367
Los detalles	367
Configuración de un servidor DNS	370
Definición de una zona primaria en el archivo named.conf	371
Definición de una zona secundaria en el archivo named.conf	372
Definición de una zona de caché en el archivo named.conf	373
Tipos de registros DNS	373
Preparación de los archivos de la base de datos BIND	378
División de los pasos individuales	379

La caja de herramientas DNS	383
host	383
dig	384
nslookup	386
whois	387
nsupdate	388
La herramienta rndc	388
Configuración de clientes DNS	389
El Resolver	390
La configuración del cliente	392
Resumen	393
▼ 17 FTP	495
La mecánica de FTP	396
Interacciones cliente / servidor	396
Obtención e instalación de vsftpd	398
La configuración de vsftpd	399
Personalización del servidor FTP	407
Resumen	411
▼ 18 Puesta en marcha de un servidor Web utilizando Apache	413
Comprensión del protocolo HTTP	414
Encabezados	414
Puertos	415
Titularidad de procesos y seguridad	416
Instalación del servidor HTTP Apache	417
Los módulos de Apache	418
Inicio y apagado de Apache	419
Inicio de Apache durante el arranque del sistema	420
Pruebas a su instalación	421
Configuración del servidor Apache	421
Creación de una sencilla página de nivel raíz	421
Archivos de configuración de Apache	422
Opciones de configuración comunes	422
Localización de problemas en Apache	427
Resumen	428
▼ 19 SMTP	429
Comprensión de SMTP	430
Detalles rudimentarios de SMTP	430
Implicaciones de seguridad	432
Instalación del servidor Postfix	433
Instalación de Postfix mediante RPM	433
Instalación de Postfix desde el código fuente	434

Configuración del servidor Postfix	436
El archivo main.cf	436
Revise su configuración	438
Inicio de operaciones del servidor	438
Revise la cola de correo	439
Vacíe la cola de correo	439
El comando newaliases	439
Asegúrese de que todo funciona	439
Resumen	440
▼ 20 POP e IMAP	441
Funcionamiento de POP e IMAP	444
Prueba del servidor POP	444
Prueba del servidor IMAP	445
Instalación del servidor UW-IMAP y POP3	446
Otros temas sobre servicios de correo	449
Seguridad SSL	449
Disponibilidad	450
Archivos de registro	451
Resumen	451
▼ 21 Secure Shell (SSH)	453
Conceptos básicos sobre criptografía de llave pública	454
Características de las llaves	456
Referencias sobre criptografía	457
Comprensión de las versiones y distribuciones de SSH	457
OpenSSH y OpenBSD	458
Otros proveedores de clientes de SSH	458
El eslabón más débil	459
Descarga, compilación e instalación de SSH desde código fuente	459
Instalación de OpenSSH mediante RPM	462
Inicio y detención del servidor	462
El archivo de configuración SSHD	463
Uso de Openssh	464
El cliente ssh	464
Creación de un túnel seguro	464
Secure Copy (scp)	468
Secure FTP (sftp)	468
Archivos utilizados por SSH	469
Resumen	470

Parte V**Servicios intranet**

▼ 22 Network File System (NFS)	473
Funcionamiento de NFS	474
Versiones de NFS	475
Consideraciones de seguridad de NFS	476
Montaje y acceso a una partición	476
Habilitación de NFS	477
Los componentes de NFS	478
Soporte del núcleo para NFS	479
Configuración de un servidor NFS	480
El archivo de configuración /etc(exports	480
Informe al proceso del servidor NFS acerca de /etc(exports	482
El comando showmount	483
Localización de fallas del lado del servidor NFS	484
Configuración de clientes de NFS	484
El comando mount	485
Montajes duros <i>vs</i> suaves	485
Montaje cruzado de disco	485
Importancia de la opción intr	488
Ajustes finos al desempeño	488
Localización de fallas del lado del cliente de NFS	489
Identificadores de archivos viciados	489
Permisos denegados	489
Configuración del cliente y del servidor NFS	490
Usos comunes de NFS	491
Resumen	492
▼ 23 Network Information Service (NIS)	493
Contenido de NIS	494
Los servidores NIS	495
Dominios	496
Configuración del servidor NIS maestro	496
Establecimiento del nombre del dominio	497
El arranque de NIS	498
Edición del archivo Makefile	498
El uso de ypinit	502
Configuración de un cliente de NIS	504
Edición del archivo /etc/yp.conf	504
Para habilitar e iniciar ypbnd	505
Edición del archivo /etc/nsswitch.conf	506
NIS trabajando	508
Pruebas de la configuración del cliente de NIS	510

Configuración de un servidor NIS secundario	510
Configuración del nombre de dominio	510
Configuración del NIS maestro para la propagación hacia servidores esclavos	511
Activación de ypinit	511
Herramientas NIS	512
Uso de NIS en archivos de configuración	513
Puesta en marcha de NIS en una red existente Una red pequeña	514
Una red segmentada	514
Redes de mayor tamaño que los edificios	515
Resumen	515
▼ 24 Samba	517
La mecánica de SMB	518
Nombres de usuarios y contraseñas	518
Contraseñas con encriptación	519
Demonio Samba	519
Instalación de Samba	520
Compilación e instalación de Samba desde el código fuente	521
Administración de Samba	523
Inicio y terminación de Samba	523
Uso de SWAT	524
Puesta en marcha de SWAT	524
Menús de SWAT	526
Creación de un recurso compartido	527
Uso de smbclient	529
Navegación por un servidor	530
Acceso a archivos remotos	530
Montaje de recursos remotos compartidos mediante Samba	532
Creación de usuarios de Samba	532
Uso de contraseñas NULL	533
Cambio de contraseñas con smbpasswd	533
Uso de Samba para autenticación contra un servidor Windows	534
Localización de fallas en Samba	535
Resumen	536
▼ 25 LDAP	537
Fundamentos de LDAP	538
El directorio LDAP	538
Modelo cliente/servidor	539
Usos de LDAP	540
Terminología de LDAP	540

OpenLDAP	541
Demonios utilizados del lado del servidor	541
Utilidades OpenLDAP	542
Instalación de OpenLDAP	542
Configuración de OpenLDAP	543
Configuración de slapd	544
Inicio y terminación de slapd	547
Configuración de clientes de OpenLDAP	548
Creación de objetos en el directorio	549
Búsqueda, consulta y modificación del directorio	550
Uso de OpenLDAP para autenticación de usuarios	552
Resumen	555
▼ 26 Servicios de impresión	557
Terminología de impresión	558
El sistema CUPS	559
Funcionamiento de CUPS	559
Instalación de CUPS	559
Configuración de CUPS	560
Agregado de impresoras	561
Impresoras locales e impresoras remotas	562
Uso de la interfaz Web	563
Para agregar una impresora desde una línea de comandos	565
Administración de rutina de CUPS	567
Configuración de la impresora predeterminada	567
Activación y desactivación de impresoras	567
Aceptación y rechazo de trabajos de impresión	567
Gestión de privilegios de impresión	568
Eliminación de impresoras	568
Gestión de impresoras mediante la interfaz Web	568
Uso de herramientas de impresión del lado del cliente	569
lpr	570
lpq	570
lprm	571
Resumen	571
▼ 27 DHCP	573
La mecánica de DHCP	574
El servidor DHCP	575
Instalación del software DHCP mediante RPM	575
Configuración del servidor DHCP	575
Un archivo dhcpcd.conf de muestra	581
Comportamiento general durante el tiempo de ejecución ..	582
El demonio del cliente de DHCP	583
Configuración del cliente de DHCP	583
Resumen	584

▼ 28 Copias de seguridad	585
Evaluación de necesidades para copias de seguridad	586
¿Cuántos datos?	586
¿Qué tipo de medios de almacenamiento?	587
Consideraciones sobre el desempeño	
de cintas magnéticas	588
¿Cuánto ancho de banda necesitará en su red?	588
¿Qué velocidad de restauración?	588
¿Cómo administrará las cintas?	589
Manejo del dispositivo de cintas magnéticas	589
Uso de mknod y scsidev para crear los archivos	
del dispositivo	590
Manipulación de dispositivos de cinta magnética	
con mt	591
Herramientas desde la línea de comandos	592
dump y restore	592
tar	597
Resumen	597
▼ Índice	599

RECONOCIMIENTOS



Estoy loco. Este libro es prueba de ello.

Entre el trabajo y otras responsabilidades, me imaginé que agregar a lo anterior la concepción del proyecto de un libro no sería demasiado malo... Gracias a Dios que contara con la colaboración de mi maravillosa esposa Heidi, incluso después de que me dijo que estaba loco. Ella es más de lo que merezco.

Antes que Heidi viniera a mi rescate, mis padres, Manjari y Jagdish Shah, fueron los directos responsables de darme las herramientas, el amor y el ánimo para no cejar en el logro de mis intereses.

Ha sido absolutamente fantástico trabajar con mi coautor, Adewale Soyinka. Sus concepciones, retroalimentación e insistencia en que siempre siguiéramos el camino correcto ayudaron a hacer de este libro algo de lo que ambos estamos orgullosos. Las personas que se encuentran detrás del escenario en Osborne también han colaborado para que este libro salga a la luz. Jane Brownlow (la persona responsable de la primera edición de este libro!), Jennifer Housh, Emily Rader y el equipo completo de editores nos ayudaron a reunir todo para que el libro resultara lo mejor posible. Agradezco de la manera más sincera que hayan tenido la paciencia para trabajar con nosotros en todos los detalles a fin de que este libro sea una realidad.

Al final, pero desde luego no menos importante, se encuentran mis amigos en NetScaler. Su incesante energía y trabajo en equipo mantuvieron mi espíritu despierto, incluso cuando estaba físicamente exhausto. Constituyen un grupo increíble para trabajar.

—Steve

La lista de las personas a las que me gustaría manifestarles mi agradecimiento es un tanto larga, y, en tal forma, intentaré crear un “incluye todo” que refleje los individuos y grupos que lo merecen. Esto sencillamente incluye a todos aquellos que alguna vez han creído en mí y me han dado una u otra oportunidad para experimentar diversos aspectos de mi vida hasta ahora. Ustedes saben quiénes son y les doy las gracias, y permanezco en deuda con ustedes para siempre.

—Wale

INTRODUCCIÓN

El 5 de octubre de 1991, Linus Torvalds envió este mensaje al grupo de noticias comp.os.minix. (Después de todo, hubo una época en la que los grupos de noticias no estaban contaminados con noticias inútiles o enojosas.)

¿No echan de menos los gratos días de minix-1.1, cuando los hombres eran más hombres y escribían sus propios controladores de dispositivos? ¿No cuentan con un buen proyecto y sólo se están muriendo por echar sus dientes en un OS (sistema operativo) que puedan intentar modificar según sus necesidades? ¿Lo encuentran frustrante cuando todo funciona sobre minix? ¿No más desvelados para hacer que funcione un programa excelente? Entonces este mensaje podría ser precisamente para ustedes :-)

Linus continuó hasta presentar la primera parte de Linux al mundo. Para su sorpresa, había liberado lo que se convirtió en uno de los sistemas operativos más populares del planeta, en segundo lugar sólo después de Windows de Microsoft. Catorce años después, toda una industria ha crecido en torno a Linux. Posibilidades las hay, es probable que usted ya lo haya usado en una forma o en otra. (¿Tiene un TiVo? Ha usado Linux.)

SUGERENCIA Para tener una visión completa de lo ingenioso del primer mensaje de Linus Torvalds acerca de Linux, busque “Linus’ first post about Linux” en Google. El primer vínculo lo conducirá a la entrada sobre Google Groups. También puede conocer más acerca de la historia de Linux en http://en.wikipedia.org/wiki/Linux_kernel.

El Linux del año pasado no es del todo semejante al Linux de hoy. Cuando los autores instalaron por primera vez Slackware en 1994, hacer trabajar los periféricos básicos fue un reto. Por ejemplo, el entorno de la GUI de Linux requirió horas de afinación. El proceso podría causar que su monitor hiciera el pavoroso sonido “clic”, lo cual significaría que sobrepasó el cronometraje de un ajuste y tendría que comprar un monitor nuevo. De manera análoga, hacer que funcionara un software descargado de Internet típicamente necesitó que el administrador depurara los problemas del código fuente. Linux puede haber sido un sistema operativo sólido, pero no era para alguien de corazón tímido.

Como contraste, en la actualidad, la mayor parte de la distribución de Linux prácticamente se instala y configura por sí misma. Casi todos los periféricos que se puedan concebir funcionan fuera de la caja. Lo que resulta irónico es que, ahora, hacer trabajar el software en *otras* plataformas UNIX representa un cierto reto porque la mayor parte del software nuevo está escrito sobre sistemas Linux y requiere la instalación de puertos. (Buena suerte si usted todavía administra Irix o HPUX.)

Este cambio trae consigo toda una industria de soporte, opciones de infraestructura e, incluso, software de clase empresarial. Después de todo, cuando los semejantes de Oracle hacen su primera emisión de un nuevo software disponible para Linux *antes de para* cualquier otro sistema operativo, usted sabe que tiene que realizar algún esfuerzo en serio.

Este libro, escrito originalmente en 1999/2000, también ha cambiado de manera significativa con los momentos que se viven. La cuarta edición representa actualizaciones significativas para reflejar el estado actual de Linux, las posibilidades de las que disponen los administradores y las realidades del mercado de poner en marcha un servidor de clase empresarial en un mundo centrado en Internet, en donde debe disponerse con facilidad de todo, desde la seguridad hasta la satisfacción de las necesidades básicas de acceso.

Este libro *no* es un relato minuciosamente detallado de por qué los otros sistemas operativos y sus usuarios son tontos por tomar esas decisiones. Linux, como cualquier tecnología, es una herramienta. Nuestro propósito es darle a usted la mejor información posible acerca de la puesta en marcha de un servidor Linux, de modo que pueda tomar la mejor decisión para la solución de sus problemas. La religión de los sistemas operativos no tiene cabida aquí.

QUIÉN DEBE LEER ESTE LIBRO

El título de este libro incluye “Guía del principiante” y, en su mayor parte, eso es cierto. Lo que el título debe decir es “Principiantes en la guía de Linux”, porque hacemos unas cuantas suposiciones acerca de usted, el lector.

En primer lugar, suponemos que ya está familiarizado con el cuidado y la alimentación de Windows en el nivel de “usuario poderoso” o mejor. Suponemos que está familiarizado con los términos necesarios para hacer funcionar una red Windows de tamaño pequeño hasta mediano. Toda experiencia con redes más grandes o tecnologías avanzadas de Windows, como Active Directory (Directorio activo), le permitirán obtener más del libro, aunque no se requiere.

Hacemos esta suposición porque nuestra intención no era la de escribir una guía para quienes desconocieran el tema por completo. Ya existen libros suficientes en el mercado que le dicen en dónde hacer clic, sin decirle por qué; no se pretende que este libro se encuentre entre esas categorías. Además, no queríamos desperdiciar el tiempo escribiendo acerca de información que creemos es de conocimiento común por los usuarios poderosos de Windows. Otras personas ya han realizado un trabajo excelente al dar a conocer esa información y no hay razón para repetir ese trabajo aquí.

Además de sus bases de Windows, suponemos que está interesado en tener más información acerca de los temas que se encuentran aquí que del material que hemos escrito por separado. Después de todo, ¡sólo hemos consumido de 30 a 40 páginas sobre temas que tienen libros enteros dedicados a ellos! Por esta razón, a través de todos los capítulos, hemos esparcido referencias hacia otros libros. Le instamos a que tome ventaja de estas recomendaciones. No importa cuán avanzado esté usted, siempre hay algo nuevo que aprender.

¿QUÉ ESTÁ EN ESTE LIBRO?

Manual de administración de Linux, está dividido en cinco partes.

Parte I: Instalación de Linux como servidor

La parte I incluye tres capítulos (capítulo 1, “Resumen técnico de las distribuciones de Linux y de Windows 2003”; capítulo 2, “Instalación de Linux en configuración de servidor”, y capítulo 3, “Instalación de software”) que le dan a usted un manejo firme acerca de lo que es Linux, la manera en que se compara éste con Windows en varias áreas clave, cómo instalar Fedora Core y, por último, cómo instalar el código fuente y el software preempacado. De manera ideal, ésta debe ser información suficiente para que usted se inicie y le ayude a encontrar paralelismo de la manera en que trabaja Linux con base en lo que conoce de Windows.

Parte II: Administración con un solo anfitrión

En la parte II se cubre el material necesario para administrar un solo sistema que se desconecta de una red. Aun cuando esto puede parecer inútil en un principio, es el fundamento sobre el cual se construyen muchos otros conceptos, y estos últimos son esenciales para entender incluso después de que un sistema se conecta a una red.

En esta parte, se tienen siete capítulos. En el capítulo 4, “Administración de los usuarios”, se cubre la información necesaria acerca de cómo agregar, eliminar y, de otra manera, administrar los usuarios. En ese capítulo, también se presentan los conceptos básicos de operación con usuarios múltiples y cómo se aplica un usuario a la seguridad de la aplicación. En el capítulo 5, “La línea de comandos”, empezamos a cubrir los aspectos básicos del trabajo con la línea de comandos de Linux, de modo que usted pueda llegar a sentirse cómodo al separarse del entorno gráfico proporcionado de modo predeterminado. Aunque es posible administrar un sistema desde el interior del escritorio gráfico, el poder más grande se obtiene al sentirse cómodo tanto con la CLI como con la GUI. (Esto también se cumple para Windows. ¿No lo cree? Abra un mensaje de comando, ejecute `netsh` e intente hacer que `netsh` lo haga en la GUI.)

Una vez que se sienta cómodo con la CLI, empiece el capítulo 6, “Inicialización y apagado”, en el cual se documenta el proceso completo de inicialización y apagado. Esto incluye el detalle necesario sobre cómo iniciar los servicios y cómo apagarlos de modo apropiado en el curso de estos ciclos, de modo que usted pueda agregar, sin dificultad y en forma confiable, nuevos servicios más adelante en el libro.

En el capítulo 7, “Sistemas de archivos”, se continúa con los aspectos básicos de los sistemas de archivo: su organización, creación y, lo que es más importante, su administración. En el capítulo 8, “Servicios centrales del sistema”, se continúa con los aspectos básicos de la operación, con cobertura de las herramientas básicas, como `xinetd`, para programar aplicaciones que se van a ejecutar en momentos especificados. `Xinetd` es el equivalente de Linux de `svchost` y `syslog` de Windows, los cuales

administran el registro cronológico (logging) para todas las aplicaciones, en un marco unificado. Se puede concebir **syslog** como una versión más flexible del Event Viewer (Visor de eventos).

Se termina esta sección con los capítulos 9, “Compilación del núcleo de Linux” y 10, “Perillas y carátulas: el sistema de archivos proc”, en los cuales se cubre el núcleo y la afinación al nivel de núcleo a través de **/proc**. En la cobertura del núcleo se documenta el proceso de compilación e instalación de su propio núcleo personalizado en Linux. Esta capacidad es uno de los puntos que les da a los administradores de Linux una cantidad extraordinaria de mucho control sobre la forma en que operan sus sistemas. La visión de la configuración y variables al nivel de núcleo, a través del sistema de archivos **/proc**, mostrado en el capítulo 10, permite a los administradores realizar una optimización fina de la operación de su núcleo en lo que equivale a una manera posiblemente mejor y más fácil que **regedit** de Windows.

Parte III: Seguridad y operación en red

En las ediciones anteriores de este libro la seguridad y la operación en red se ubicaron al final de él. Se hizo esto porque en esa época las únicas extensiones reales al libro que se cubrieron fueron conceptos avanzados de operación en red que no se aplican a la mayor parte de los administradores. Las cosas han cambiado de manera significativa en los últimos años.

Con la importancia creciente de la seguridad en Internet así como aspectos de cumplimiento con Sarbanes Oxley y HIPAA, el uso de Linux en situaciones de seguridad ha crecido en forma drástica. Por tanto, decidimos trasladar la cobertura de estos temas antes de presentar los servicios basados en la red, los cuales podrían ser objeto de ataques en ésta.

Arrancamos esta sección con el capítulo 11, “TCP/IP para administradores de sistemas”, en el cual se proporciona un panorama detallado de TCP/IP en el contexto de lo que los administradores de sistemas necesitan saber. En el capítulo se dan muchos detalles sobre la manera de usar las herramientas de detección de fallas, como **tcpdump**, para capturar paquetes y leerlos de nuevo, así como un análisis paso a paso de la manera como trabajan las conexiones TCP. Estas herramientas deben proporcionar las herramientas para que usted sea capaz de investigar en forma efectiva las peculiaridades de la red.

En el capítulo 12, “Configuración de la red”, se regresa a los conceptos de administración y se enfoca en la configuración básica de la red. Esto incluye la estructuración de las direcciones IP, las entradas de router e, incluso, la estructuración de múltiples direcciones IP. En el capítulo 13, “Configuración del firewall de Linux”, nos extendemos más allá de los conceptos básicos, yendo hacia los conceptos avanzados de la operación en red y mostrando cómo puede usted construir un firewall basado en Linux.

En los capítulos 14, “Seguridad local” y 15, “Seguridad en la red”, se analizan con detalle los aspectos de seguridad de la red y del sistema. Estos incluyen aspectos específicos de Linux, así como sugerencias y trucos acerca de la seguridad en general, de modo que usted pueda configurar mejor su sistema y protegerlo contra ataques.

Parte IV: Servicios de Internet

El resto del libro está dividido en dos partes distintas: Servicios de Internet y de intranet. Definimos los servicios de Internet como aquellos que puede usted considerar que se ejecutan en un sistema Linux expuesto de manera directa a Internet. Ejemplos de esto incluyen los servicios de la Web y DNS.

Iniciamos esta sección con el capítulo 16, “DNS”. En esta sección, cubrimos la información que usted necesita conocer para instalar, configurar y administrar un servidor DNS. Además de los detalles reales para hacer funcionar un servidor DNS, suministramos fundamentos detallados

sobre cómo trabaja DNS así como varias sugerencias para la localización de fallas, trucos y herramientas. De DNS, nos movemos hacia el capítulo 17, “FTP”, y cubrimos la instalación y cuidado de los servidores FTP. Como en el capítulo sobre DNS, también incluimos los fundamentos sobre el propio protocolo FTP y algunas notas sobre su evolución.

En el capítulo 18, “Puesta en marcha de un servidor Web utilizando Apache”, nos movemos sobre lo que puede ser considerado uno de los usos más populares hoy en día de Linux: hacer funcionar un servidor Web con el servidor Web Apache. En este capítulo, cubrimos la información necesaria para instalar, configurar y administrar el servidor Web de Apache; le damos los detalles adicionales para la estructuración y configuración del funcionamiento como anfitrión virtual.

En los capítulos 19, “SMTP” y 20, “POP e IMAP”, nos sumergimos en el correo electrónico a través de la estructuración y configuración de los servidores SMTP, POP e IMAP. Cubrimos la información necesaria para configurar los tres, así como también mostramos cómo interactúan entre sí, se pueden probar a través de CLI y Telnet, y se pueden usar en entornos heterogéneos. Lo que el lector puede hallar un poco diferente acerca de este libro en relación con otros sobre Linux es que hemos elegido cubrir el servidor Postfix SMTP, en lugar del servidor clásico Sendmail, debido a que suministra un servidor más flexible con un mejor récord de seguridad.

Terminamos la parte con el capítulo 21, “Secure Shell (SSH)”. La estructuración y administración del servicio SSH es un requisito para cualquier servidor que el lector estuture, sin importar la función de correo del mismo.

Parte V: Servicios intranet

Definimos los servicios de intranet como aquellos que típicamente funcionan detrás de un firewall sólo para usuarios internos. Incluso en este entorno, Linux tiene mucho que ofrecer. Empezamos por ver NFS en el capítulo 22, “Network File System (NFS)”. NFS ha estado por allí durante cerca de 20 años y ha evolucionado y crecido para ajustarse muy bien a las necesidades de sus usuarios. En este capítulo, cubrimos las capacidades de servidor NFS de Linux, incluyendo cómo estructurar tanto los clientes como los servidores, así como la localización de fallas. Desde NFS nos movemos hasta NIS, en el capítulo 23, “Network Information Service (NIS)”. Típicamente, NIS se despliega al lado de los servidores NFS para suministrar un servicio central de nombramiento para todos los usuarios dentro de una red. Ponemos atención especial en aumentar la escala de los temas y mostrar cómo puede usted hacer que NIS funcione en un entorno con una base grande de usuarios.

En el capítulo 24, “Samba”, se continúa con la idea de compartir discos y recursos con cobertura del servicio Samba. Si usan Samba, los administradores pueden compartir discos y medios de impresión con usuarios de Windows, sin tener que instalar algún software de cliente especial. Por consiguiente, Linux puede convertirse en un servidor eficaz, capaz de soportar sistemas UNIX en los que se ejecute NFS, así como sistemas Windows en los que se ejecute Samba.

Volvemos a visitar los servicios de directorio en el capítulo 25, “LDAP”, con cobertura de LDAP y la manera en que los administradores pueden usar este servicio estándar con el fin de suministrar directorios de usuarios para sistemas operativos múltiples, así como directorios de correo tanto para sistemas Windows como UNIX.

En el capítulo 26, “Servicios de impresión”, hacemos un recorrido por el subsistema de impresión de Linux. El subsistema de impresión en combinación con Samba permite a los administradores soportar impresión fluida proveniente de escritorios Windows. El resultado es una manera poderosa de centralizar las opciones de impresión para usuarios de Linux, Windows e, incluso, Mac OS X en un solo servidor.

En el capítulo 27, “DHCP”, se cubre otro uso común de los sistemas Linux: los servidores DHCP. En este capítulo, cubrimos la manera de desplegar el servidor ISC DHCP, lo cual ofrece un poderoso arreglo de características y controles de acceso que no se exponen de manera tradicional en las herramientas de administración DHCP con base gráfica.

Finalizamos con el capítulo 28, “Copias de seguridad”. Se puede argumentar que los respaldos constituyen una de las piezas más críticas de la administración. Linux viene con dos medios comunes de proporcionar copias de seguridad que son fáciles de usar y que se pueden usar de buena gana en las unidades de cinta. Cubrimos las dos y explicamos cómo se pueden usar como parte de un programa de respaldo. Además de la mecánica de los respaldos, analizamos el diseño general de éstos y cómo puede usted optimizar su propio sistema.

Actualizaciones y retroalimentación

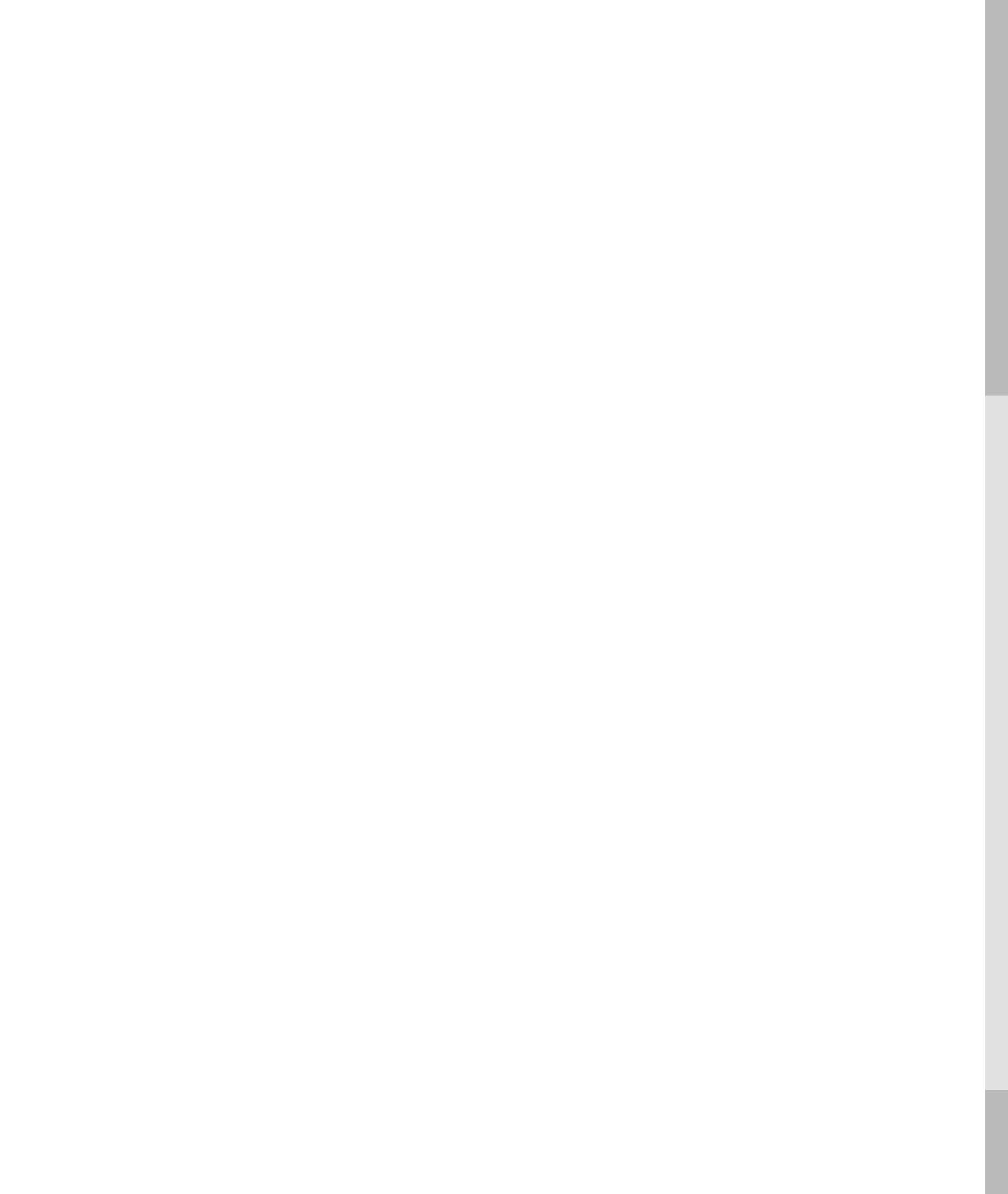
Aun cuando esperamos que se publique un libro sin errores, esto no siempre es posible. El lector puede hallar una lista de erratas de este libro publicada en <http://www.planetoid.org/linux>. Si encuentra cualesquiera errores, daremos la bienvenida a sus propuestas de actualizaciones de las erratas.

También daremos la bienvenida a su retroalimentación y comentarios. Por desgracia, nuestras tareas diarias nos impiden dar respuesta a preguntas detalladas, de modo que si está buscando ayuda sobre un tema específico, puede hallar como una mejor selección una de las muchas comunidades en línea. (Un gran lugar para empezar es <http://www.linux.org/>.) Sin embargo, si tiene dos céntimos que quiera compartir acerca del libro, le daremos la bienvenida a sus ideas. Nos las puede enviar por correo electrónico a linuxadmin@planetoid.org.

PARTE I



Instalación de Linux
como servidor



CAPÍTULO 1



Resumen técnico de
las distribuciones
de Linux y de
Windows 2003

Linux ha impactado la tendencia principal. Un rápido recorrido por cualquier tienda importante de su localidad que se especialice en electrónica y computadoras hará ver lo siguiente: ¡las ofertas de software incluyen versiones en caja de varias distribuciones de Linux! Lo que fue sólo un juguete para un hacker hace un par de años ha crecido en forma tremenda y se conoce por su desempeño estable y rápido del servidor. Si se necesitaran más pruebas, sólo observe una pregunta común que ahora se hace a los CTO (jefes técnicos ejecutivos) de las compañías de Fortune 500. La pregunta es: "¿Cuál es su estrategia Linux?"

Con los entornos KDE y GNOME, Linux también está abriendo caminos hacia el interior del mercado de los escritorios de Windows. En este capítulo, daremos un vistazo a las diferencias técnicas entre Linux y el Windows.NET Server (la plataforma que es probable esté usted considerando reemplazar con Linux). En este capítulo, también se explica la licencia GNU (no UNIX de GNU), lo cual puede ayudarle a comprender por qué gran parte de Linux es como es.

APRENDIZAJE ACERCA DEL SISTEMA OPERATIVO LINUX

Por lo común, la gente entiende que Linux es un paquete completo de herramientas de desarrollo, editores, GUI, herramientas para operación en red, etc. De modo más formal, ese tipo de paquetes se llaman *distribuciones*. Lo más probable es que el lector haya escuchado acerca de las distribuciones de Linux nombradas Red Hat, Fedora Core, Mandrake y suSE, las cuales se han dado a conocer con profusión y han sido adquiridas por miles de instalaciones. Las distribuciones no comerciales de Linux, como Debian, son menos conocidas y no han alcanzado la misma escala de popularidad, pero allí están, y en uso activo por parte de sus respectivas (y muy activas) comunidades.

Lo que resulta interesante acerca de todas las distribuciones de Linux es que casi todas las herramientas con las cuales se embarcan no fueron escritas por las propias compañías. En lugar de ello, otras personas han lanzado sus programas con licencias, permitiendo su redistribución con el código fuente. De manera general, también se cuenta con estas herramientas en otras variantes de UNIX, y también se está empezando a disponer de algunas de ellas en Windows. Quienes se encargan de la distribución sencillamente las colocan en un paquete conveniente que se instala con facilidad. (Algunos distribuidores también desarrollan herramientas con valor agregado que hacen que su distribución sea más fácil de administrar o que sea compatible con más hardware pero, en general, el software que embarcan lo escriben otros.)

De modo que si consideramos que una distribución es todo lo que usted necesita para Linux, entonces ¿qué es exactamente Linux? El propio Linux es el corazón del sistema operativo: el *núcleo*. El núcleo es el programa que actúa como Jefe de Operaciones. Es el responsable de arrancar y detener otros programas (como los editores), manejar las solicitudes de memoria, proporcionar acceso a discos y administrar las conexiones de la red. La lista completa de actividades del núcleo podría ser con facilidad el contenido de un capítulo y, de hecho, se han escrito varios libros en los que se documentan las funciones internas del núcleo.

El núcleo se conoce como un programa no trivial. También es el que pone el Linux en todas aquellas que sean distribuciones de Linux. En esencia, en todas las distribuciones se usa el mismo núcleo y, como consecuencia, el comportamiento fundamental de todas las distribuciones de Linux es el mismo.

Diferencias del núcleo

Cada compañía que vende una distribución de Linux de inmediato le dirá que su núcleo es mejor que los demás. ¿Cómo puede una compañía hacer esta afirmación? La respuesta viene del hecho de que en la actualidad cada compañía mantiene su propio juego de parches. Con el fin de tener la seguridad de que los núcleos permanecen en su mayor parte en sync, la mayoría en realidad adopta parches que se ponen en el árbol de Linux (como se publica en <http://www.kernel.org>). La única diferencia es que, por lo general, los vendedores no ponen en circulación toda versión del núcleo que se publica en kernel.org. En lugar de ello, toman una base, aplican sobre ella sus parches personalizados, hacen funcionar el núcleo a través de un sencillo proceso QA y, en seguida, lo llevan a producción. Esto ayuda a las organizaciones a tener confianza en que sus núcleos tienen suficiente respaldo, mitigando de esta manera cualquier riesgo que se haya percibido acerca de ejecutar sistemas operativos de fuente abierta.

La única excepción a esta regla gira en torno a aspectos de seguridad. Si se encuentra algo relacionado con la seguridad con un núcleo Linux, los vendedores están prestos a adoptar los parches necesarios para arreglar el problema de inmediato. En el transcurso de un corto periodo (por lo común, en menos de 24 horas), se hace una nueva emisión del núcleo, de modo que los administradores puedan tener la confianza de que sus instalaciones están seguras. Es de agradecer que las explosiones contra el propio núcleo sean raras.

De manera que si cada vendedor mantiene su propio juego de parches, ¿qué es exactamente lo que está parchando? Esta respuesta varía de vendedor a vendedor, dependiendo del mercado hacia el cual cada uno de ellos se enfoca. Por ejemplo, Red Hat se enfoca en gran parte en el suministro de confiabilidad y eficiencia sólida del grado empresarial para servidores de aplicaciones. Ésta puede ser diferente a la misión del equipo de Fedora Core, el cual se encuentra más interesado en intentar con rapidez nuevas tecnologías, e incluso más diferente que la del enfoque de un vendedor que está intentando reunir un sistema Linux orientado al escritorio.

Lo que separa una distribución de la siguiente son las herramientas de valor agregado que vienen con cada una. Por ejemplo, Red Hat incluye una herramienta muy útil que hace que la configuración de la interfaz gráfica sea una tarea muy directa. Preguntar “¿cuál distribución es mejor?” es casi como preguntar “¿cuál es mejor, la Coca o la Pepsi?” Casi tienen los mismos ingredientes básicos, agua carbonatada, cafeína y jarabe de maíz con alto contenido de fructosa, lo que da el efecto de mitigar la sed y traer como consecuencia una sensación de placer producido por la cafeína y el azúcar. Al final de cuentas, es una cuestión de necesidades: ¿Necesita usted un apoyo comercial? ¿Su vendedor de aplicaciones le recomendó una aplicación por encima de otra? ¿La infraestructura de actualización del paquete se adapta mejor a su estilo de administración del sitio que la de otra distribución? Cuando usted revise sus necesidades, encontrará que es probable que exista una distribución que se ajuste con exactitud a ellas.

¿QUÉ HAY EN TORNO DEL SOFTWARE LIBRE Y DE GNU?

A principios de la década de 1980. Richard Stallman inició un movimiento dentro de la industria del software. Predicó (y todavía lo hace) que el software debe ser libre. Note que con el término libre Stallman no se refiere al precio, sino más bien libre en el sentido de libertad. Esto significó embarcar no sólo un producto, sino también todo el código fuente.

La política de Stallman era, un tanto irónicamente, un regreso a la computación clásica, cuando el software se compartía con libertad entre los aficionados a las computadoras pequeñas y los vendedores de macro y minicomputadoras lo daban como parte del hardware (no fue sino hasta finales de la década de 1960 que IBM consideró vender el software de aplicación. Durante la década de 1950 y la mayoría de la de 1960, consideraron el software sencillamente como una herramienta para hacer posibles las ventas del hardware).

Este regreso a la apertura fue una brusca separación de las convenciones de principios de la década de 1980, de vender el software preempacado, pero el concepto de Stallman concordaba con las distribuciones iniciales de UNIX de Bell Labs. Los primeros sistemas UNIX sí contenían el código fuente completo. Sin embargo, a finales de la década de 1970, era típico que se eliminara el código fuente de las distribuciones de UNIX y sólo se podía adquirir mediante el pago de grandes cantidades de dinero a AT&T (ahora SBC). La Berkeley Software Distribution (BSD) mantuvo una versión libre, pero su contraparte comercial, BSDi, tuvo que enfrentar muchas demandas de AT&T hasta que se pudo probar que nada en el núcleo de BSD era de AT&T.

La idea de regalar el código fuente es sencilla: los usuarios del software nunca deben ser forzados a tratar con un desarrollador que podría o no dar apoyo a las intenciones que ese usuario tuviera para el software. El usuario nunca debe tener que esperar que se publiquen los arreglos de errores. Lo que es más importante, es normal que el código desarrollado bajo el escrutinio de otros programadores sea de mayor calidad que el escrito detrás de puertas cerradas. Sin embargo, el mayor beneficio del software libre proviene de los propios usuarios: si necesitaran una nueva característica, pueden agregarla al programa y, a continuación, contribuir con ella de regreso a la fuente, de modo que todos los demás puedan resultar beneficiados de ello.

Siguiendo esta línea de razonamiento, ha surgido un deseo de dar a la luz pública un sistema completo semejante al UNIX, libre de las restricciones de las licencias. Por supuesto, antes de que usted pueda construir cualquier sistema operativo, necesita construir herramientas, y ésta es la manera como nació el proyecto GNU.

NOTA GNU representa no UNIX de GNU; los acrónimos recursivos son parte del humor de los hackers. Si usted no entiende por qué es gracioso, no se preocupe; todavía se encuentra dentro de la mayoría.

¿Qué es la GNU Public License?

Algo muy importante que surgió del proyecto GNU ha sido la *GNU Public License (GPL)*. Esta licencia expresa a detalle que el software que se está emitiendo es libre y nadie puede quitar nunca estas libertades. Es aceptable tomar el software y revenderlo, incluso para obtener una utilidad; sin embargo, en esta reventa, el vendedor debe emitir el código fuente completo, incluyendo cualesquiera cambios. En virtud de que el paquete revendido permanece bajo la GPL, el paquete se puede distribuir en forma libre y todavía volverse a revender por alguien más con el fin de obtener una utilidad. De importancia primordial es la cláusula de responsabilidad: los programadores no son responsables de cualesquier daños causados por su software.

Se debe hacer notar que la GPL no es la única licencia usada por los desarrolladores de software libre (aunque se puede argumentar que es la más popular). Otras licencias, como la BSD y la Apache, tienen cláusulas semejantes de responsabilidad, pero difieren en los términos de su redistribución. Por ejemplo, la licencia BSD permite a la gente hacer cambios al código y embarcar estos cambios, sin tener que revelar el código agregado (la GPL requeriría que se embarcara el código agregado). Para obtener más información acerca de otras licencias de fuente abierta, consulte <http://www.opensource.org>.

Nota histórica al pie

Hace un par de años, Red Hat inició una oferta comercial de su en otro tiempo producto libre (Red Hat Linux). La emisión comercial fue la serie Red Hat Enterprise Linux (RHEL). Debido a que el fundamento de RHEL es GPL, los individuos interesados en mantener una versión libre de la distribución de Red Hat han sido capaces de hacerlo. Además, como una extensión para la comunidad, Red Hat creó el Fedora Core Project, el cual se considera como las bases de prueba para un nuevo software antes de que sea adoptado por el equipo de RHEL. El Fedora Core Project se distribuye en forma libre y se puede descargar de <http://fedora.redhat.com>.

Ventajas del software libre

Si la GPL parece una mala idea desde el punto de vista del comercialismo, considere el surgimiento reciente de los exitosos paquetes de freeware; éstos son indicativos de un sistema que en realidad funciona. Este éxito ha evolucionado por dos razones. En primer lugar, como se mencionó con anterioridad, con mucho los errores en el propio código es más probable que sean captados y arreglados con rapidez bajo los ojos observadores de los colegas. En segundo, bajo el sistema GPL, los programadores pueden distribuir el código sin el miedo de ser demandados. Sin esa protección, jamás nadie haría público su código.

Por supuesto, este concepto da por sentada la pregunta de por qué cualquiera haría público su software en forma gratuita. La mayor parte de los proyectos no arrancan con todas sus características, como piezas pulidas de trabajo. Pueden iniciar su vida como un rápido resultado de un hacker, para resolver un problema específico que molesta al programador. Como una resolución rápida e imperfecta, el código no tiene valor de venta; pero se convierte en una herramienta útil cuando se comparte con otros y, como consecuencia, es mejorado por estos últimos, quienes tienen problemas y necesidades semejantes. Otros usuarios del programa empiezan a acrecentarlo con características que necesitan, y estas adiciones hacen su viaje de regreso al programa original. De este modo, el proyecto evoluciona como resultado del esfuerzo de un grupo y llega el momento en que alcanza su pleno refinamiento. Este programa pulido puede contener colaboraciones de posiblemente cientos, si no es que miles, de programadores que han añadido pequeñas piezas aquí y allá. De hecho, es posible que el código del autor original resulte poco evidente.

Existe otra razón para el éxito del software con licencia general. Cualquier administrador de proyectos que ha trabajado sobre software comercial sabe que el costo *real* del software en desarrollo no se encuentra en esa fase de desarrollo. En realidad está en el costo de venta, mercadeo, soporte, documentación, empaque y embarque de ese software. A un programador, que consume un tiempo de fin de semana como hacker con el fin de arreglar un problema con un programa diminuto e imperfecto, le puede faltar interés, tiempo y dinero para hacer que este resultado se convierta en un producto reeditable.

Cuando Linus Torvalds sacó Linux a la luz pública en 1991, lo hizo bajo la GPL. Como resultado de su carta abierta, Linux ha tenido un número notable de colaboradores y analizadores. Esta participación ha hecho que Linux sea muy fuerte y rico en características. El propio Torvalds estima que desde el núcleo v.2.2.0, las colaboraciones de él representan sólo el 5% de la base total del código.

Ya que cualquiera puede tomar el núcleo Linux (y otros programas de soporte), volver a empaquetarlos y revenderlos, algunas personas han hecho dinero con él. En tanto que estas emisiones individuales emitan el código fuente completo del núcleo junto con sus propios paquetes y mientras estos paquetes estén protegidos bajo la GPL, pueden ser revendidos por otras personas, con otros nombres, con el fin obtener una utilidad.

Al final de cuentas, lo que hace que un paquete de una persona sea más valioso que el de otra consiste en las características de valor agregado, los canales de soporte y la documentación. Incluso IBM puede estar de acuerdo con esto; es como han hecho la mayor parte de su dinero desde 1930 hasta 1970 y ahora, al final de la década de 1990 y principios de los años 2000, con los IBM Global Services. El dinero no está en el producto; se encuentra en los servicios que van con él.

COMPRENSIÓN DE LAS DIFERENCIAS ENTRE WINDOWS Y LINUX

Como el lector podría imaginar, las diferencias entre Windows de Microsoft y el sistema operativo Linux no se pueden discutir por completo en los confines de esta sección. A lo largo de estos capítulos, tema por tema, examinaremos los contrastes específicos entre los dos sistemas. En algunos capítulos, el lector encontrará que no hacemos comparaciones porque, en realidad, no existe una diferencia importante.

Pero antes de tocar los detalles, tomemos un momento para discutir las diferencias arquitectónicas principales entre los dos sistemas operativos.

Usuarios únicos en comparación con usuarios múltiples y en comparación con usuarios de redes

Windows se diseñó según la visión de Bill Gates, cofundador de Microsoft, “una computadora, un escritorio, un usuario”. En beneficio de la discusión, llamaremos a esta filosofía de *usuario único*. En esta disposición, dos personas no pueden trabajar en paralelo ejecutando (por ejemplo) Microsoft Word en la misma máquina, al mismo tiempo. (Por otra parte, juno podría preguntarse lo sabio de hacer esto con un programa abrumadoramente pesado como Word! Usted puede comprar Windows y ejecutar lo que se conoce como Terminal Server, pero esto requiere un enorme poder de computación y grandes cantidades de dinero en la adquisición de licencias. Por supuesto, con Linux se ejecutará bastante bien con sólo casi cualquier hardware.

Linux pide prestada su filosofía a UNIX. Cuando se desarrolló originalmente UNIX en los Bell Labs, a principios de la década de 1970, existió en una computadora PDP-7 que necesitaba ser compartida por todo un departamento. Esto requirió un diseño que permitiera que se les diera acceso (log in) a *múltiples usuarios* a la máquina central, al mismo tiempo. Varias personas pudieron editar documentos, compilar programas y realizar otro trabajo exactamente al mismo tiempo. El sistema operativo de la máquina central tenía a su cargo los detalles del “compartimiento”, de modo que parecía que cada usuario tenía un sistema separado. Esta tradición de múltiples usuarios continuó hasta hoy en día, también en otras versiones de UNIX. Y desde el nacimiento de Linux, a principios de la década de 1990, ha soportado la disposición de múltiples usuarios.

NOTA La mayor parte de las personas cree que con la llegada de Windows 95 se inventó el término de "tareas múltiples". ¡UNIX ha tenido esta capacidad desde 1969! El lector puede sentirse seguro de que los conceptos puestos en Linux han tenido muchos años para desarrollarse y probarse.

En la actualidad, la implementación más común de una estructura de múltiples usuarios es soportar *servidores*: sistemas dedicados a ejecutar programas grandes para que los usen muchos clientes. Cada miembro de un departamento puede tener una estación de trabajo más pequeña sobre el escritorio, con poder suficiente para realizar el trabajo cotidiano. Cuando necesiten hacer algo que requiera más poder de la CPU o más memoria, pueden ejecutar la operación en el servidor.

El lector puede argumentar: "¡Pero, oiga! ¡Windows puede permitir a la gente descargar el trabajo intensivo desde el punto de vista computacional hacia una sola máquina! ¡Sólo considere el SQL Server!" Bien, esa posición sólo es correcta a medias. Tanto Linux como Windows de hecho son capaces de suministrar servicios, como las bases de datos, sobre la red. Podemos dar el nombre de *usuarios de red* a los usuarios de esta disposición, ya que, en realidad, nunca se les da acceso al servidor sino, más bien, envían solicitudes a éste. El servidor realiza el trabajo y, a continuación, envía los resultados de regreso al usuario a través de la red. La trampa es que, en este caso, debe escribirse específicamente una aplicación para que realice ese tipo de obligaciones servidor / cliente. Dentro de Linux, un usuario puede ejecutar en el servidor cualquier programa permitido por el administrador del sistema, sin tener que volver a diseñar ese programa. La mayor parte de los usuarios encuentran que es un beneficio significativo la capacidad de ejecutar programas arbitrarios en otras máquinas.

El núcleo monolítico y el micronúcleo

En los sistemas operativos, se tienen dos formas de núcleos. Usted tiene un núcleo monolítico que suministra todos los servicios que necesitan las aplicaciones de los usuarios, y, enseguida, tiene el micronúcleo, un conjunto central pequeño de servicios y otros módulos que ejecutan otras funciones.

En su mayor parte, en Linux se adopta la arquitectura del núcleo monolítico; éste maneja todo lo que se refiera al hardware y a las llamadas del sistema. Windows funciona por completo en un diseño de micronúcleo. El núcleo proporciona un conjunto pequeño de servicios y, a continuación, se interconecta con otros servicios ejecutivos que suministran la administración del proceso, administración de ENT/SAL y otros servicios. Todavía tiene que probarse cuál metodología es en verdad la mejor.

Separación de la GUI y el núcleo

Tomando una entrada del concepto de diseño de Macintosh, los desarrolladores de Windows integraron la interfaz gráfica del usuario (GUI) con el sistema operativo central. Sencillamente, uno no existe sin el otro. El beneficio para este íntimo acoplamiento del sistema operativo y la interfaz del usuario es la firmeza en el aspecto del sistema.

Aun cuando Microsoft no impone reglas tan estrictas como Apple con respecto al aspecto de las aplicaciones, la mayor parte de los desarrolladores tienden a adherirse a una apariencia y sensación básicas entre las aplicaciones. Una razón para que esto sea peligroso es que ahora se le permite al controlador de la tarjeta de video que funcione en lo que se conoce como "Ring 0" en una arquitectura x86 típica. Ring 0 es un mecanismo de protección; sólo los procesos privilegiados pueden ejecutarse a este nivel y lo normal es que los procesos del usuario se ejecuten en Ring 3. Ya que se permite que la tarjeta de video funcione en Ring 0, esta tarjeta podría desempeñarse mal (¡y lo hace!), lo cual puede desplomar el sistema completo.

Por otra parte, Linux (como UNIX en general) ha mantenido separados los dos elementos: la interfaz del usuario y el sistema operativo. La interfaz X Window System se ejecuta como una aplicación del nivel usuario, lo cual la hace más estable. Si la GUI (la cual es muy compleja tanto para Windows como para Linux) falla, la parte central de Linux no se cae con ella. El proceso sencillamente se desploma y usted llega a una ventana terminal. X Window System también difiere de la GUI de Windows en que no es una interfaz completa del usuario; sólo define cómo deben extraerse y manipularse los objetos básicos sobre la pantalla.

La característica más significativa de X Window System es su capacidad para presentar ventanas de uno a otro lado de la red y sobre otra pantalla de la estación de trabajo. Esto permite a un usuario situado en el anfitrión A obtener el acceso (log in) al anfitrión B, ejecutar una aplicación en el anfitrión B y tener toda la salida dirigida de regreso al anfitrión A. Es posible que dos personas que obtengan el acceso a la misma máquina, ejecuten al mismo tiempo un equivalente de Linux de Word de Microsoft (como OpenOffice).

Además de X Window System central, se necesita un administrador de ventanas con el fin de crear un entorno útil. Las distribuciones de Linux vienen con varios administradores de ventanas e incluyen soporte para GNOME y KDE, los cuales también se encuentran en otras variantes de UNIX. Si usted se encuentra preocupado respecto a la velocidad, puede considerar los administradores de ventanas WindowMaker y FVWM. Éstos podrían no contar con todo el fulgor de KDE o GNOME pero en realidad son rápidos. Cuando se fijan como predeterminados, tanto GNOME como KDE ofrecen un entorno que es amigable incluso para el usuario no frecuente de Windows.

De modo que ¿cuál es mejor —Windows o Linux— y por qué? Eso depende de lo que usted está tratando de hacer. El entorno integrado que proporciona Windows es conveniente y menos complejo que Linux, pero le falta la característica de X Window System que permite a las aplicaciones presentar sus ventanas de uno a otro lado de la red en otra estación de trabajo. La GUI de Windows es uniforme pero no se puede apagar, en tanto que el X Window System no tiene que estar ejecutándose (y consumiendo memoria valiosa) en un servidor.

Network Neighborhood

El mecanismo nativo de la gente de Windows para compartir discos en los servidores o entre sí es a través de Network Neighborhood. En un escenario típico, los usuarios *se vinculan* a un compartimiento y tienen al sistema asignando a éste una letra de unidad. Como resultado, la separación entre cliente y servidor resulta clara. El único problema con este método de compartimiento de datos es que está más orientado a la gente que orientado a la tecnología: la gente tiene que saber cuáles servidores contienen cuáles datos.

Con Windows, también ha aparecido una nueva característica que ha pedido prestada a UNIX: el *montaje*. En la terminología de Windows, esto se conoce como *puntos de repartición*. Ésta es la capacidad de montar una unidad de CD-ROM en un directorio en su unidad C. Esto puede parecer un poco extraño, pero a medida que use Linux comprenderá que ésta es la *única* manera de montar trabajos. Lo asombroso es que en Windows no puede montar compartimientos de la red de esta manera. Tiene que trazar un mapa de un compartimiento de la red hasta una letra de unidad.

Linux, mediante el uso de Network File System (NFS), ha soportado el concepto de montaje desde su comienzo. De hecho, Linux Automounter puede montar y desmontar dinámicamente las particiones de acuerdo con las necesidades.

Un ejemplo común de montaje de particiones bajo Linux comprende directorios iniciales montados. Los directorios iniciales del usuario residen en un servidor y el cliente monta los directorios en el momento de arranque (automáticamente). De modo que **/home** existe en el cliente, pero **/home/username** existe en el servidor.

Bajo el NFS de Linux, los usuarios nunca tienen que conocer los nombres del servidor ni las trayectorias de los directorios y esa ignorancia le da felicidad a usted. No más preguntas acerca de a cuál servidor hay que conectarse. Todavía mejor, los usuarios no necesitan saber cuándo surge la necesidad de cambiar la configuración del servidor. Con Linux, usted puede cambiar los nombres de los servidores y ajustar esta información en los sistemas del lado del cliente sin anunciarlo o tener que volver a educar a los usuarios. Cualquiera que alguna vez ha tenido que volver a orientar a los usuarios acerca de las nuevas disposiciones del servidor tiene conciencia de las repercusiones que se pueden presentar.

La impresión funciona en gran parte de la misma manera. Con Linux, las impresoras reciben nombres que son independientes del nombre real del anfitrión de la impresora. (Esto tiene una importancia especial si la impresora no habla TCP/IP.) Los clientes apuntan a un servidor de impresión cuyo nombre no se puede cambiar sin autorización administrativa. Los ajustes no pueden ser cambiados sin el conocimiento de usted. El servidor de impresión entonces puede dirigir hacia otro lado todas las solicitudes de impresión, según se necesite. La interfaz uniforme de Linux recorrerá un largo camino hacia la mejora de lo que puede ser una disposición caótica de las impresoras en la instalación de usted. Esto también significa que no tiene que instalar controladores de impresión en varios lugares.

NOTA Si pretende usar Linux para dar servicio a clientes de Windows/NT/98 a través del paquete de Samba, todavía tendrá que lidiar con la notificación a los usuarios con respecto a los compartimientos del servidor y las asignaciones de las impresoras. En el capítulo 24, puede leer más acerca de Samba.

El Registry en comparación con los archivos de texto

Piense en el Registry de Windows como la última base de datos de configuración: miles sobre miles de entradas, muy pocas de las cuales están documentadas por completo, algunas localizadas en servidores y algunas en clientes.

“¿Qué? ¿Dijo usted que su Registry se corrompió?” <risas maniáticas> “Bien, sí, podemos tratar de restablecerlo a partir de los respaldos de anoche, pero entonces Excel empieza a actuar de manera rara y el técnico (quien cobra 50 dólares sólo por responder el teléfono) dijo que se necesita una reinstalación....”

En otras palabras, el sistema Registry de Windows es, en el mejor de los casos, muy difícil de administrar. Aunque, en teoría, es una buena idea, la mayor parte de la gente que tiene tratos con él no sale de la batalla sin una cicatriz o dos.

Linux no cuenta con un registro. Esto es tanto una bendición como una maldición. La bendición es que, en su mayor parte, los archivos de configuración se guardan como una serie de archivos de texto (piense en los archivos .INI de Windows anteriores a los días de Registry). Esta estructura significa que usted puede editar los archivos de configuración con el uso del editor de textos que elija, en lugar de las herramientas como **regedit**. En muchos casos, también significa que puede hacer comentarios con toda libertad sobre esos archivos de configuración, de modo que, dentro de seis meses, no olvidará por qué estructuró algo de una manera en particular. Con la mayor parte de las herramientas que vienen con Linux, los archivos de configuración existen en el directorio **/etc** o en uno de sus subdirectorios.

Más sobre archivos de configuración

Un efecto lateral interesante de tener la existencia de archivos de configuración como una serie de archivos de texto es que la configuración de estos archivos se puede automatizar. Esto es muy útil en situaciones en donde se necesita desplegar un número grande de estaciones de trabajo o se agregan con frecuencia nuevas estaciones de este tipo. Usted puede estructurar con facilidad Linux en un disco CD-ROM que sencillamente monta un compartimiento NFS e inicia una instalación personalizada. Aun cuando el lector puede hacer esto en Windows, la manera soportada comprende el uso de diferencias binarias llamadas sys-diffs, y scripts automatizados de instalación (archivos .ini). La otra manera es usar herramientas para clonar, como Norton Ghost y duplicar la instalación.

El problema con esto es que cada caja de Windows tiene un Security Identifier (SID) asociado con ella y éste se almacena a lo largo de todo Registry. Se pueden hacer instalaciones automatizadas de Windows; sólo que requiere mucho más esfuerzo.

La maldición de la disposición sin registro es que no se cuenta con una manera estándar de escribir los archivos de configuración. Cada aplicación o servidor puede tener su propio formato. Hoy en día, muchas aplicaciones llegan empaquetadas con herramientas de configuración basadas en la GUI para aligerar algunos de estos problemas. De modo que el lector puede realizar con facilidad una estructuración básica y, a continuación, editar en forma manual el archivo de configuración cuando necesite hacer ajustes más complejos.

En realidad, tener archivos de texto para mantener la información de la configuración por lo común resulta ser un método eficiente. Una vez fijados, rara vez necesitan cambiarse; incluso de ser necesario el cambio, son archivos de texto directos y, por consiguiente, fáciles de visualizar cuando es necesario. Incluso de mayor utilidad es que resulta fácil escribir scripts para leer los mismos archivos de configuración y modificar en consecuencia su comportamiento. Esto es útil en especial al automatizar las operaciones de mantenimiento del servidor, lo cual es crucial en un sitio grande con muchos servidores.

Dominios y Active Directory

Si ha estado usando Windows durante un tiempo suficiente, es posible que recuerde el modelo controlador de dominios de Windows NT. Si siente punzadas de miedo que le recorren el cuerpo al leer la última oración, puede ser que todavía esté sufriendo la neurosis de guerra de tener que mantener Primary Domain Controllers (PDC), Backup Domain Controllers (BDC) y su sincronización.

Microsoft, temiendo la rebelión de los administradores de todo el mundo, hizo a un lado el modelo de Windows NT y creó el Active Directory (AD). La idea detrás del AD era sencilla: suministrar un almacén para cualquier tipo de datos administrativos, sean permisos de acceso (logins) a usuarios, información de grupos o, incluso, tan solo números telefónicos, y administrar la autenticación y la autorización para un dominio. También se cambió el modelo de sincronización de los dominios para seguir jerarquía del estilo DNS, la que ha probado ser bastante más confiable. También se eliminó NTLM a favor de Kerberos. (Note que el AD todavía es compatible con NTLM.)

Mientras se ejecuta, **dcpromo** puede que no sea la idea de nadie de una tarde divertida, es fácil ver que el AD funciona bastante bien.

Linux no tiene una autentificación/autorización íntimamente acopladas y un modelo de almacenamiento de datos de la manera que los tiene Windows con Active Directory. En lugar de ello, Linux utiliza un modelo de abstracción que permite tipos múltiples de almacenamiento y esquemas de autentificación, para trabajar sin modificación alguna de las aplicaciones. Esto se realiza a través de la infraestructura de Password Authentication Models (PAM) y las bibliotecas de resolución de nombres que proporcionan un medio estándar de buscar la información de los grupos para las aplicaciones y una manera flexible de almacenar esa información de los grupos usando diversos esquemas.

Para los administradores que acuden a Linux, esta capa de abstracción puede parecer peculiar al principio. Sin embargo, considere que puede usar cualquier cosa, desde archivos planos, pasando por NIS, hasta LDAP o Kerberos, para la autentificación. Esto significa que puede elegir el sistema que mejor le funcione. Por ejemplo, si cuenta con una estructura UNIX existente, en la que se usa NIS, puede hacer sencillamente que sus sistemas Linux se inserten en eso. Por otra parte, si tiene una infraestructura AD existente, puede usar PAM con Samba o LDAP, para autenticar contra el dominio. ¿Usa Kerberos? No hay problema. Y, por supuesto, puede elegir que su Linux no interactúe con ningún sistema externo de autentificación. Además de ser capaz de ligarse a múltiples sistemas de autentificación, Linux puede usar con facilidad diversas herramientas, como OpenLDAP, para mantener también disponible la información del directorio.

OTRAS REFERENCIAS

Si el lector tiene interés en estar bajo la capucha de la revolución tecnológica (y siempre resulta útil saber cómo funcionan las cosas), se recomiendan los textos siguientes:

- ▼ *Computer: A History of the Information Machine*, por Martin Campbell-Kelly y William Aspray (HarperCollins, 1997)
- ▲ *A Quarter Century of Unix*, por Peter Salus (Addison-Wesley, 1994)

En ninguno de estos textos se discute específicamente Linux. De hecho, en *A Quarter Century of Unix* se relata la historia de Linux hasta el punto en donde el sistema sólo se estaba convirtiendo en serio jugador. Peter Salus escribe una discusión interesante de por qué, en primer lugar, Linus Torvalds se vio en la necesidad de crear Linux.

Para obtener la exclusiva sobre el propio Linux, arranque con la página de inicio de Linux en <http://www.linux.org>.

CAPÍTULO 2



Instalación de Linux
en configuración
de servidor

Un atributo clave en el éxito reciente de Linux es la notable mejora en las herramientas de instalación. Lo que una vez fue un proceso ligeramente aterrador hace muchos años ahora se ha vuelto casi trivial. Lo que es aún mejor, se cuenta con muchas maneras para instalar el software; los medios ópticos (CD/DVD-ROM) ya no son la única elección (aun cuando todavía son los más comunes). Las instalaciones a través de la red también forman parte de la lista predeterminada de posibilidades y pueden constituirse en una ayuda maravillosa cuando se instala un gran número de anfitriones.

La mayor parte de las configuraciones predeterminadas en donde se instala Linux ya son capaces de convertirse en servidores. Esto se debe a una decisión de diseño desafortunada y un tanto ingenua: ¡ser designada como servidor significa que la máquina sirve para todo! Desde los servicios de disco hasta las impresoras, hasta el correo, hasta las noticias...todo se activa desde el arranque. La práctica apropiada exige que una máquina llamada servidor debe dedicarse a efectuar sólo una o dos tareas específicas. Cualesquiera otros servicios instalados e irrelevantes sencillamente ocupan memoria y constituyen un obstáculo para el rendimiento. En este capítulo, se discute el proceso de instalación según pertenece a los servidores y sus funciones dedicadas.

HARDWARE Y CONSIDERACIONES AMBIENTALES

Como con cualquier sistema operativo, antes de adentrarse en el arranque del proceso de instalación, debe determinar cuáles configuraciones del hardware funcionarían. Cada vendedor comercial publica una lista de compatibilidades del hardware (HCL) y la pone a disposición de los interesados en su sitio Web. Por ejemplo, la HCL de Red Hat se encuentra en <http://hardware.redhat.com/hcl> (se puede suponer con seguridad que la HCL de Fedora Core es semejante a la de Red Hat), la base de datos HCL de SuSE se puede hallar en <http://hardwaredb.suse.de> y se puede concentrar una HCL más genérica para la mayor parte de las orientaciones de Linux en <http://www.tldp.org/HOWTO/Hardware-HOWTO>.

Estos sitios proporcionan un buen punto de referencia de arranque cuando usted tiene dudas referentes a una pieza de hardware en particular. Sin embargo, tenga presente que, en todo el mundo, diariamente se están produciendo con profusión nuevos controladores de dispositivos Linux y que ningún sitio puede mantener el paso del desarrollo en la comunidad de fuente abierta. En general, las configuraciones más populares basadas en Intel y en AMD funcionan sin dificultad.

Una sugerencia general que se aplica a todos los sistemas operativos (OS) es evitar configuraciones de hardware y software que estén de moda. Aunque parecen ser en realidad impresionantes, no han tenido el proceso de maduración por el que algo del hardware un tanto más antiguo ha pasado. Para los servidores, esto no suele ser una necesidad, ya que no es necesario que un servidor tenga las herramientas más recientes y más grandes, como tarjetas de video y de sonido de lujo. Después de todo, la meta principal de usted es suministrar un servidor estable y que se encuentre disponible para sus usuarios.

DISEÑO DEL SERVIDOR

Cuando un sistema se convierte en un servidor, su estabilidad, disponibilidad y rendimiento se convierten en un aspecto significativo. Estos tres factores suelen mejorarse con la compra de más

hardware, lo cual es desafortunado. Resulta una vergüenza pagar miles de dólares adicionales para tener un sistema capaz de desempeñarse en las tres áreas cuando usted pudo haber extraído el nivel deseado de rendimiento del hardware existente con un poco de afinación. Con Linux esto no es difícil. Incluso todavía mejor, las ganancias son sobresalientes.

La decisión más significativa respecto al diseño que usted debe tomar al administrar la configuración de un servidor no es técnica sino administrativa. Debe diseñar un servidor que *no sea amigable* para los usuarios casuales. Esto significa que no tenga herramientas bonitas de multimedia, soporte de tarjeta de sonido ni navegadores lujosos de la Web (cuando sea posible). De hecho, debe ser una regla que el uso casual de un servidor esté estrictamente prohibido.

Otro aspecto importante del diseño de un servidor es asegurarse que tiene un buen entorno. Como administrador de un sistema, debe garantizar la seguridad física de sus servidores manteniéndolos en una sala separada, con candado y llave (o lo que sea equivalente). El único acceso a los servidores para el personal no administrativo debe ser a través de la red. La propia sala de los servidores debe estar bien ventilada y mantenerse fría. Es posible que ocurra un accidente si no hay el medio ambiente adecuado. Los sistemas que se sobrecalentan y los usuarios entrometidos que piensan que saben cómo arreglar los problemas pueden ser un peligro tan grande para la estabilidad del servidor como el software malo (y se puede argumentar que lo son todavía más).

Una vez que el sistema se encuentra en un lugar seguro, también es crucial la instalación del respaldo de baterías. La energía eléctrica de respaldo tiene dos finalidades clave:

- ▼ Mantener el sistema funcionando durante una falla de energía eléctrica de modo que se pueda apagar con eficiencia, evitando de esta manera la pérdida o daño de datos
- ▲ Garantizar que los picos y caídas de tensión, así como otros ruidos no interfieran con el buen funcionamiento de su sistema

Enseguida, se dan algunas sugerencias específicas que puede hacer para mejorar el rendimiento de su servidor:

- ▼ Saque ventaja del hecho de que la interfaz gráfica del usuario no esté acoplada al sistema operativo central y evite arrancar X Window System (la GUI de Linux) a menos que alguien necesite sentarse ante una consola y ejecutar una aplicación. Después de todo, como cualquier otra aplicación, X Window System requiere memoria y tiempo de la CPU para funcionar, los cuales es mejor aplicarlos en los procesos más esenciales del servidor.
- Determine qué funciones va a realizar el servidor y desactive todas las demás. Las funciones no usadas no sólo son un desperdicio de memoria y tiempo de la CPU, sino sólo son otro asunto con el que necesita tratar en el frente de seguridad.
- ▲ A diferencia de algunos otros sistemas operativos, Linux permite tomar y elegir las características deseadas en el núcleo (en el capítulo 9, el lector aprenderá acerca de este proceso). El núcleo predeterminado ya se encontrará razonablemente afinado, de modo que no tendrá que preocuparse acerca de ello. Pero si en realidad necesita cambiar una característica o actualizar el núcleo, sea quisquilloso acerca de lo que agrega. Asegúrese de que en realidad necesita una característica, antes de añadirla.

NOTA Es posible que el lector oiga una vieja recomendación acerca de que si recopila su núcleo, podrá hacer más eficaces los recursos de su sistema. Esto ya no es del todo cierto; las otras razones para recomilar su núcleo podrían ser actualizar o agregar soporte para un nuevo dispositivo.

Tiempo útil

Todo este parloteo referente a tener cuidado con los servidores y asegurarse de que no se desploman a causa de cosas absurdas, contiene toda una filosofía de hace largo tiempo de UNIX: *el tiempo útil es bueno; más tiempo útil es mejor*.

El comando **uptime** de UNIX (Linux) le informa al usuario cuánto tiempo ha estado funcionando el sistema desde su último arranque, a cuántos usuarios se ha permitido el acceso hasta el momento y cuánta carga está experimentando el sistema. Las dos últimas son medidas útiles necesarias para mantener día a día el buen funcionamiento del sistema y en la planificación a largo plazo (por ejemplo, a últimas fechas, la carga del servidor ha estado elevada, de modo que puede ser el momento de comprar un servidor más rápido / más grande / mejor).

Pero lo más importante es cuánto tiempo ha estado funcionando el sistema desde su último reinicio. Los tiempos útiles muy largos son un signo de cuidado apropiado, mantenimiento y, desde un punto de vista práctico, de estabilidad del sistema. A menudo encontrará a los administradores de UNIX alardeando respecto a los tiempos útiles de su servidor de la misma manera que oye a los aficionados de los automóviles alardear en relación con los caballos de potencia. Ésta también es la razón por la que escuchará a los administradores de UNIX maldiciendo los cambios al sistema (sin importar el sistema operativo) que requieren que tenga lugar un reinicio. Puede ser que ahora niegue usted que eso le preocupe, pero en un plazo de seis meses probablemente le gritará a quienquiera que reinicie el sistema de manera innecesaria. No se moleste en tratar de explicar este fenómeno a alguien que no sea administrador, porque sólo lo mirarán en forma extraña. Sólo dentro de su corazón usted sabrá que su tiempo útil es mejor que el de ellos.

ASPECTOS DE LA INICIALIZACIÓN DUAL DEL SISTEMA

Si usted es novato para Linux, puede ser que no esté listo para comprometerse con un sistema completo cuando sólo desea una unidad de prueba. Todas las distribuciones de Linux se pueden instalar sólo sobre ciertas particiones de su disco duro dejando en tanto las demás solas. Lo típico es que esto signifique permitir que Windows de Microsoft coexista con Linux.

Debido a que nos estamos enfocando sobre las instalaciones de servidores, no cubriremos los detalles de construcción de un sistema de inicialización dual; sin embargo, cualquiera con un poco de experiencia en la creación de particiones en un disco debe ser capaz de figurarse esto. Si está teniendo dificultades, puede ser que desee consultar la guía de instalación que viene con la distribución que haya adquirido.

Algunos consejos rápidos: si, en la actualidad, una partición de Windows 95/98 consume un disco duro completo, como la unidad C, puede usar la herramienta **fips** para la repartición del disco. Sencillamente, desfragmente y, a continuación, ejecute **fips.exe**. Si está usando Windows NT/2000/XP con NTFS y ya tiene asignado todo el disco con datos en cada partición, puede ser que tenga que mover a mano un poco los datos para liberar una partición. En virtud de su complejidad, es ligeramente más difícil volver a asignar los tamaños en una partición formateada

con NTFS. No obstante, todavía es muy posible. Algunas de las distribuciones de Linux más recientes incluso le ofrecerán el cambio automático de los tamaños de su partición NTFS, durante la instalación del OS.

NOTA Desde la perspectiva de la flexibilidad, NTFS no suena como algo bueno, pero en realidad lo es. Si tiene que ejecutar Windows NT o 2000, use NTFS.

Puede ser que encuentre que el uso de una herramienta comercial, como PartitionMagic, es especialmente útil, debido a que ofrece soporte para NTFS, FAT32 y FAT normal, así como para un gran número de otros tipos de sistemas de archivos. También, su interfaz del usuario es, de manera significativa, más buena que **fips**.

MÉTODOS DE INSTALACIÓN

Con la conectividad y velocidad mejoradas tanto de las redes de zona local como de las conexiones a Internet, se está volviendo cada vez más popular la posibilidad de realizar las instalaciones sobre la red, en lugar de usar un CD-ROM local.

Las instalaciones basadas en la red son muy útiles en especial cuando se tiene necesidad de tender Linux sobre un gran número de sistemas. Este método proporciona un procedimiento rápido de instalación en el cual se pueden instalar muchos sistemas al mismo tiempo.

Dependiendo de la distribución de Linux en particular y la infraestructura de red que ya se encuentre en el lugar, se pueden diseñar instalaciones basadas en la red en torno a varios protocolos. A continuación se da una lista de algunos de los protocolos más populares sobre los cuales se realizan las instalaciones basadas en la red:

- ▼ **FTP** Éste es uno de los métodos más antiguos para realizar instalaciones de redes.
- **HTTP** El árbol de instalación se obtiene desde un servidor web.
- **NFS** El árbol de distribución se comparte/exporta en un servidor NFS
- ▲ **SMB** Este método es más o menos nuevo y no lo soportan todas las distribuciones. El árbol de instalación se puede compartir en un servidor Samba o desde una caja de Windows.

El otro método, más típico, de instalación es a través del uso de medios ópticos surtidos por el vendedor. Todas las distribuciones comerciales de Linux tienen juegos empaquetados de marca Linux que contienen los medios de instalación. También suelen hacer imágenes en CD/DVD-ROM (ISO) del OS del que disponen en sus sitios FTP y/o HTTP. Las distribuciones que no ponen a disposición sus ISO suelen tener una versión reducida del OS que ponen a disposición en un árbol repositorio en su sitio.

En este capítulo, realizaremos una instalación de la clase de servidor usando una imagen DVD que se quemó hacia un DVD. Por supuesto, una vez que usted haya pasado por el proceso de instalación a partir de un medio óptico (CD/DVD-ROM), encontrará que las instalaciones basadas en la red son muy directas. Una nota lateral referente a las instalaciones automatizadas es que las del tipo de servidor no son muy adecuadas para la automatización, debido a que cada servidor suele tener una tarea única; por consiguiente, cada servidor tendrá una configuración un poco diferente. Por ejemplo, un servidor dedicado a manejar información de registro cronológico

(logging) que se envía sobre la red va a tener una estructura de particiones especialmente grandes para los directorios apropiados de este tipo de registro, en comparación con servidor de archivos que no realiza este registro por sí mismo (la excepción obvia es para las granjas de servidores, en donde usted tiene números grandes de servidores replicados. Pero incluso esas instalaciones tienen sus matices que requieren atención para los detalles específicos para la instalación).

INSTALACIÓN DE FEDORA CORE LINUX

En esta sección, instalará la versión 4 de Fedora Core de Linux en un sistema autónomo. Tendremos un enfoque liberal para el proceso, instalando todas las herramientas que es posible sean pertinentes para las operaciones del servidor. En capítulos posteriores, se explica la finalidad de cada subsistema y se ayuda al lector a determinar cuáles en realidad necesita conservar.

NOTA No se preocupe si decide instalar una distribución que no sea Fedora; por suerte, la mayor parte de los conceptos se trasladan entre las diversas distribuciones. Algunos instaladores sólo son más bonitos que otros.

Requisitos previos del proyecto

En primer lugar, necesita descargar los ISO para Fedora Core 4 que instalará. La página web del proyecto de Fedora tiene una lista de varios espejos localizados por todo el mundo. Por supuesto, usted debe elegir el espejo que le quede más cercano desde el punto de vista geográfico. La lista de espejos se puede encontrar en <http://fedora.redhat.com/download/mirrors.html>.

La imagen DVD usada para esta instalación se descargó de <ftp://download.fedora.redhat.com/pub/fedora/linux/core/4/i386/iso/FC4-i386-DVD.iso>.

El paso siguiente es quemar el ISO en un medio apropiado. En este caso, necesitamos quemar el ISO en un DVD en blanco. Use su programa quemador favorito de CD/DVD con el fin de quemar la imagen. Recuerde que el archivo que descargó ya es una imagen exacta de un medio DVD y, por consiguiente, debe quemarse como tal. La mayor parte de los programas quemadores de CD/DVD tienen una opción para crear un CD o un DVD a partir de una imagen.

Si quemara el archivo que descargó como si fuera un archivo común de datos, finalizará con un solo archivo en la raíz de su DVD-ROM. Esto no es lo que usted desea.

El sistema que está usted instalando debe contar con un controlador DVD-ROM.

NOTA Las imágenes de instalación de Fedora también se encuentran como un juego de cuatro imágenes CD-ROM. Puede realizar la instalación usando los cuatro CD-ROM, pero hemos decidido efectuar la instalación usando un DVD-ROM, en su mayor parte por conveniencia. El uso de un solo DVD le ayuda a evitar que tenga que extraer CD a la mitad de la instalación, debido a que todos los archivos requeridos ya se encuentran en un solo DVD, en oposición a varios CD, y también porque se reducen las posibilidades de tener un medio malo de instalación (es decir, existe una probabilidad más alta de tener un CD malo en cuatro que tener un DVD malo en uno).

Empecemos el proceso de instalación.

Realización de la instalación

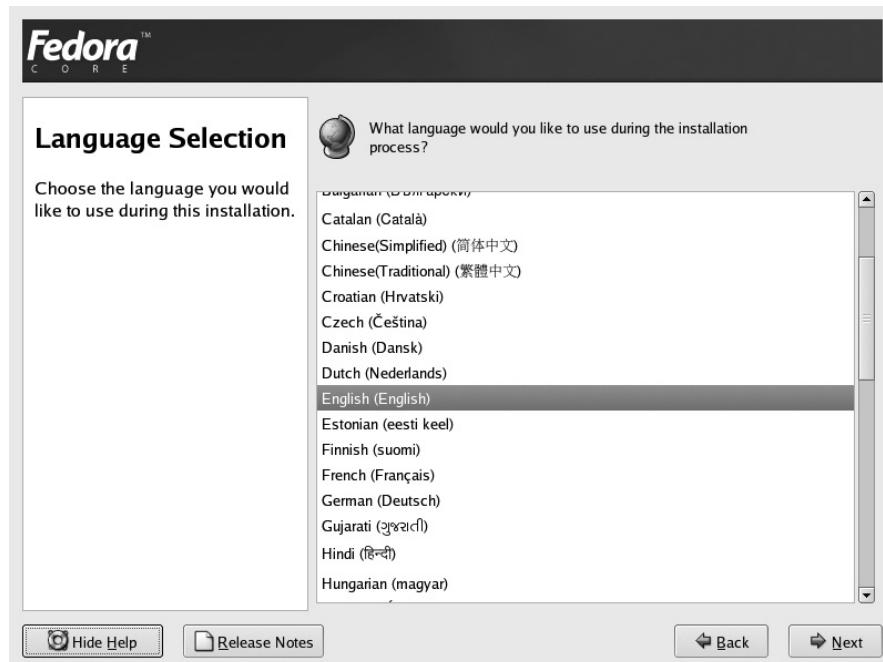
1. Para iniciar el proceso de instalación, arranque el DVD-ROM. Éste se le presentará con una pantalla de desviación (vea la ilustración) que le presenta el Fedora Core. En la parte inferior de la pantalla verá un mensaje que dice

boot:



2. Si no presiona alguna tecla, el mensaje se suspende en forma automática y empieza el proceso de instalación. Usted puede presionar **ENTER** para iniciar el proceso de inmediato.
3. En la pantalla CD Found (CD encontrado), presione **ENTER** para probar / verificar sus medios de instalación. Note que la prueba realizada aquí no siempre es perfecta. Pero en la mayoría de las veces, cuando funciona, puede ahorrarle la molestia de iniciar la instalación sólo para darse cuenta a la mitad del trayecto que el instalador abortará debido a la presencia de un disco malo.
Presione **ENTER** una vez más en la pantalla Media Check (comprobación de medios) para empezar la prueba.
4. Después de que la comprobación de medios se ejecuta por completo, debe tener una pantalla Media Check Result (Resultado de la comprobación de medios) que le informa **PASS** (Pasa). En este punto, es seguro seleccionar **OK** para continuar con la instalación.
Si no tiene algunos otros medios de instalación por probar, seleccione **Continue (Continuar)** en la pantalla siguiente.

5. Seleccione Next (Siguiente) en la pantalla Welcome To Fedora Core (Bienvenido a Fedora Core).
6. Seleccione en esta pantalla el idioma que desea usar para realizar la instalación (vea ilustración). La interfaz funciona de manera muy semejante a cualquier otra interfaz del estilo de Windows. Sencillamente coloque el puntero en su selección y haga clic. Cuando esté listo, haga clic en el botón Next en la parte inferior derecha de la pantalla. En nuestro sistema de muestra se selecciona English (Inglés).



7. Seleccione su tipo de teclado. Esta pantalla siguiente permite seleccionar el tipo de disposición para el teclado. En la división derecha del cuadro de diálogo se tiene una lista de las diversas disposiciones que se soportan.

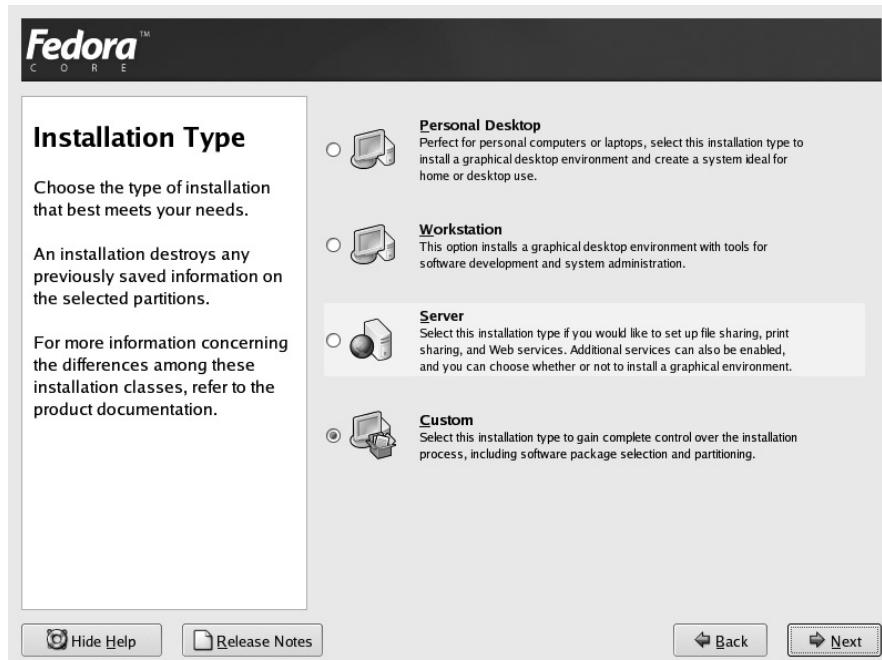
La opción más común será la disposición U.S. English (Inglés de Estados Unidos). Seleccione U.S. English y haga clic en Next para continuar.

NOTA Despues de que haga clic en Next en este punto, el instalador rápidamente buscará en su disco duro cualesquiera instalaciones existentes de Linux. Si se encuentra alguna, se pediría con una pantalla diferente que realice una actualización o una reinstalación del OS encontrado. Si, por el contrario, está instalando en un nuevo disco duro de marca, no se le presentará una pantalla de ese tipo.

Tipo de instalación

La pantalla de tipo de instalación se presenta con un conjunto de clases preconfiguradas de clases (tipos) de instalación. Las opciones son Personal Desktop (Escritorio personal), Workstation (Estación de trabajo), Server (Servidor) y Custom (Personal). Las diferentes clases comprenden paquetes y opciones que el equipo de desarrolladores de Fedora piensan que deben constituir un cierto tipo de instalación. Aun cuando hemos afirmado con anterioridad que vamos a instalar Linux en una configuración del tipo de servidor, no seleccionaremos la opción preconfigurada de Server por dos razones: una, queremos tener tanto control como sea posible sobre la manera en que se construya el sistema y dos, suponemos que usted es libre pensador y no dejará que un grupo de desarrolladores y/o autores le digan qué hacer.

1. Seleccione el botón custom (de acción excluyente).
2. Haga clic en Next, como se muestra enseguida:



Estructuración de las particiones del disco

Es probable que esta parte de la instalación sea la que la mayoría de los usuarios nuevos de Linux encuentre más difícil. Esto se debe a las convenciones diferentes acerca de los nombres que se usan en Linux. No es necesario que sea así: todo lo que exige es un ligero desplazamiento de la mente. También debe tener presente que *una partición es una partición es una partición* en Linux o en Windows.

Lo que sigue es un panorama general rápido del esquema de las particiones que estará empleando para esta instalación. En este panorama, también se dan las particiones equivalentes en el mundo de Windows:

- ▼ / La partición raíz/volumen se identifica por una diagonal hacia la derecha (/). Todos los directorios se adjuntan (montan) a este directorio padre. Es equivalente a la unidad del sistema (C:\) en Windows.
- /boot (carga inicial) Esta partición/volumen contiene casi todo lo requerido para el proceso de inicialización. Almacena los datos que se usan antes de que el núcleo empiece a ejecutar los programas del usuario. Lo equivalente de esto en Windows es lo que se conoce como la partición del sistema (*no* la partición de la carga inicial).
- /usr En ésta residirán todos los archivos de programas [semejante a C:\Program Files (Archivos de programas) en Windows].
- /home (de inicio) En ella estarán los directorios de inicio de todos (suponiendo que este servidor los alojará). Esto es útil para evitar que los usuarios consuman un disco entero y dejen otros componentes críticos sin espacio (como los archivos de registro cronológico). Este directorio es sinónimo al C:\Documents and Settings\ (documentos y ajustes) en Windows XP/200x.
- /var Ésta es donde en general se almacenan los registros cronológicos (logs) del sistema/eventos. Debido a que los archivos de estos registros tienden a crecer con rapidez y pueden ser afectados por usuarios externos (por ejemplo, individuos que visitan un sitio Web), es importante almacenar los registros cronológicos en una partición separada, de modo que nadie pueda llevar a cabo un ataque de negación del servicio al generar suficientes entradas de registro como para llenar todo el disco. En general, los registros cronológicos se almacenan en el directorio C:\WINDOWS\system32\config\ en Windows.
- /tmp En ella se colocan los archivos temporales. Debido a que este directorio se diseña de modo que cualquier usuario pueda escribir en él (semejante al directorio C:\Temp bajo Windows), es necesario que usted se asegure de que usuarios arbitrarios no abusen de ello y llenen el disco completo. Esto se garantiza manteniéndolo en una partición separada.
- ▲ Swap (Intercambio) Ésta es donde se almacena el archivo de memoria virtual. Éste no es un sistema de archivos accesible a los usuarios. Aun cuando Linux (y también otras orientaciones de UNIX) pueden usar un archivo de disco normal para mantener la memoria virtual, a la manera en que Windows lo hace, el lector encontrará que tenerla en su propia partición mejora el rendimiento. Es típico que usted querrá configurar su archivo swap para duplicar la memoria física que está en su sistema. Esto se conoce como el archivo de paginación (*paging file*) en Windows.

Cada una de estas particiones se monta en el momento de inicialización. El proceso de montaje hace que se disponga de esa partición como si sólo fuera otro directorio en el sistema. Por ejemplo, el directorio raíz (/) estará en la primera partición (raíz). En el directorio raíz, existirá un subdirectorrio llamado /usr, pero nada tendrá en él. Entonces se puede montar una partición sepa-

rada tal que ir al directorio `/usr` permitirá ver el contenido de las particiones recientemente montadas. Cuando se montan todas las particiones, aparecen como un árbol unificado de directorios, en lugar de como unidades separadas; el software de instalación no diferencia una partición de otra. De todo lo que encarga es a cuál directorio va cada archivo. Como resultado, el proceso de instalación distribuye en forma automática sus archivos de uno a otro lado de todas las particiones montadas, en tanto que estas últimas representen partes diferentes del árbol de directorios en donde suelen colocarse los archivos.

La herramienta para partir Disk Druid (Druida de disco) (vea el paso 3 en la lista próxima) fue desarrollada por Red Hat como una manera fácil de crear particiones y asociarlas a los directorios según los cuales serán montadas. Al iniciar Disk Druid, verá todas las particiones existentes en su disco, si las hay. Cada entrada de partición mostrará la información siguiente:

- ▼ **Device (Dispositivo)** Linux asocia cada partición con un dispositivo separado. Para la finalidad de esta instalación, sólo necesita saber que bajo los discos IDE, cada dispositivo empieza con `/dev/hdXY`, en donde X es *a* para un maestro IDE en la primera cadena, *b* para un esclavo IDE en la primera cadena, *c* para un maestro IDE en la segunda cadena o *d* para un esclavo IDE en la segunda cadena, y en donde Y es el número de partición del disco. Por ejemplo, `/dev/hda1` es la primera partición en la cadena primaria, disco primario. SCSI sigue la misma idea básica, excepto que en lugar de iniciar con `/dev/hd`, cada partición se inicia con `/dev/sd` y sigue el formato `/dev/sdXY`, en donde X es una letra que representa un unidad física única (*a* es para SCSI ID 1, *b* es para SCSI ID 2, y así sucesivamente). La Y representa el número de partición. Por tanto, `/dev/sdb4` es la cuarta partición en el disco SCSI con ID 2. El sistema es un poco más complejo que el de Windows, pero la ubicación de cada partición es explícita; no más conjeturas: “¿A cuál dispositivo físico corresponde la unidad E?”
- **Mount point (Punto de montaje)** La ubicación en donde se monta la partición.
- **Type (Tipo)** Este campo muestra el tipo de partición (por ejemplo, ext2, ext3, swap o vfat).
- **Format (Formato)** Este campo indica si se formateará o no la partición.
- **Size (MB) (Tamaño)** En este campo se muestra el tamaño de la partición (en MB).
- **Start (Inicio)** En este campo se muestra el cilindro sobre su disco duro en donde principia la partición.
- ▲ **End (Fin)** En este campo se muestra el cilindro sobre su disco duro en donde finaliza la partición.

En beneficio de la sencillez, sólo usará algunas de las fronteras del disco descritas con anterioridad para su instalación. Además, también dejará algo de espacio libre (espacio sin parti-

ciones) con las que podemos jugar en un capítulo posterior (capítulo 7). Usted tallará su disco duro en

/boot partition

/ partition

SWAP partition

/home partition

/tmp partition

FREE SPACE / UN-PARTITIONED AREA (Espacio libre / zona sin particiones)

El sistema muestra que donde se está realizando esta instalación tiene un disco duro de 10GB. El lector usará los tamaños que se dan a continuación como directrices acerca de cómo asignar los diversos tamaños para cada partición/volumen. Por supuesto, debe ajustar los tamaños sugeridos para adecuarse al tamaño total del disco que esté usando.

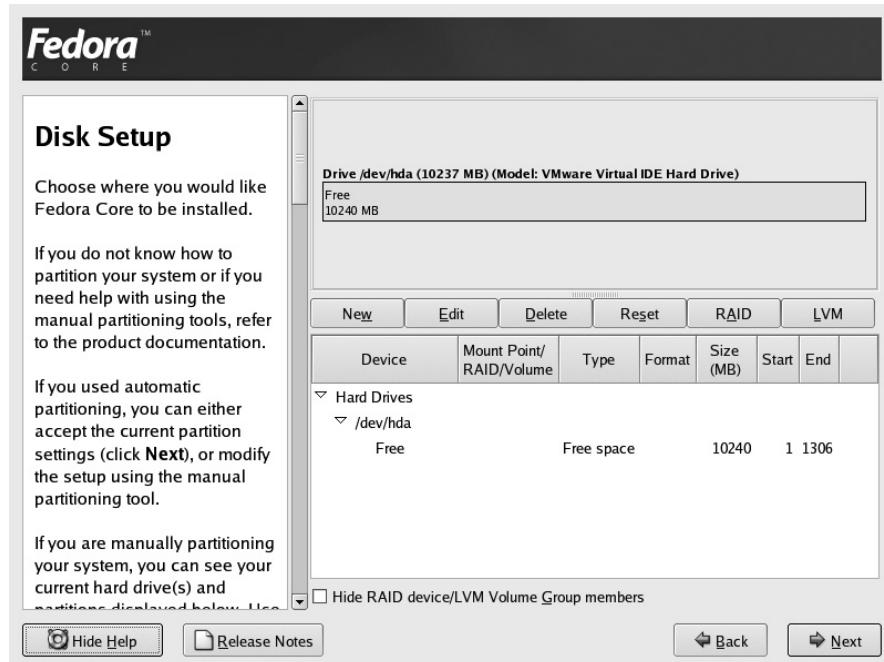
Punto de montaje	Tamaño
/boot	200MB
/	5GB
SWAP	512MB
/home	3GB
/tmp	512MB
FREE SPACE	~ 512MB

NOTA No se puede crear la partición /boot en un tipo de partición Logical Volume Management (LVM) (Administración de volumen lógico). El cargador del boot de Fedora no puede leer las particiones del tipo LVM. Esto es cierto en el momento en que se está escribiendo esto, pero puede cambiar en el futuro.

Ahora que cuenta con los fundamentos sobre particiones bajo Linux, regresemos al propio proceso de instalación:

1. Seleccione la opción Manually Partition With Disk Druid (Partición en forma manual con druida de disco) y haga clic en Next.
2. Si está realizando la instalación en un disco nuevo de marca, podría tener un cuadro de diálogo de Warning (Advertencia) que le indicará que inicialice la unidad. Seleccione Yes (Sí) cuando se le pida.

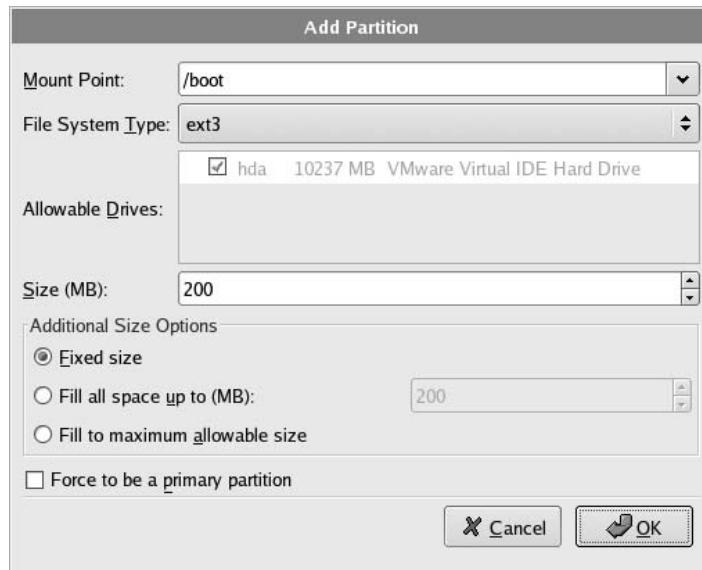
3. Enseguida, se le presentará la pantalla Disk Setup (Estructuración del disco), como se muestra a continuación:



4. Haga clic en New (Nuevo). Aparece el cuadro de diálogo Add Partition (Aregar partición); complétela con la información que sigue para los campos correspondientes:

Mount Point	/boot
File System Type (Tipo de sistema de archivos)	ext3
Allowable Drives (Unidades admisibles)	hda
Size (MB)	200
Additional Size Options (Opciones adicionales de tamaño)	Fixed Size (Tamaño fijo)
Force to be a primary partition (Forzar a ser una partición primaria)	Deje sin marca de verificación

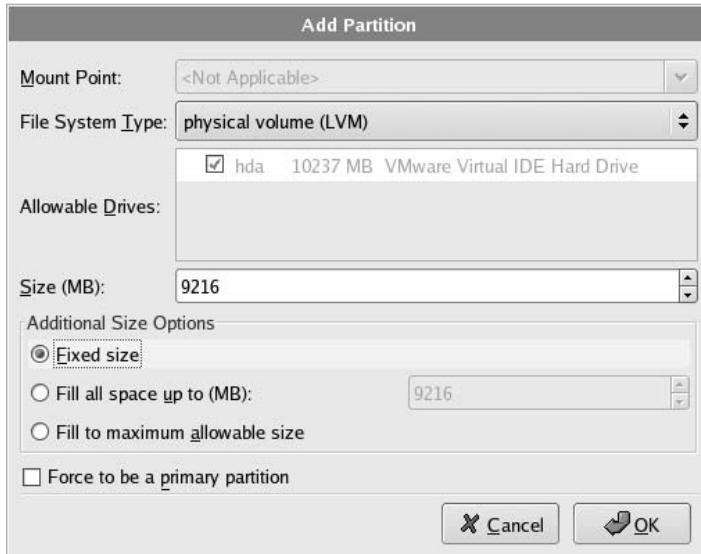
El cuadro de diálogo completado debe semejarse al que se muestra enseguida. Haga clic en el botón OK una vez que haya terminado.



5. El lector creará los recipientes / (root), /home, /tmp y swap en una partición del tipo LVM. Por lo tanto, primero necesitará crear el volumen físico padre. Haga clic en New. Aparece el cuadro de diálogo Add Partition. Se creará el volumen físico con la información que sigue:

Mount Point	Deje este campo en blanco
File System Type	physical volume (LVM) (volumen físico)
Allowable Drives	hda
Size (MB)	9216 (aproximadamente 9.0GB)
Additional Size Options	Fixed Size
Force to be a primary partition	Deje sin marca de verificación

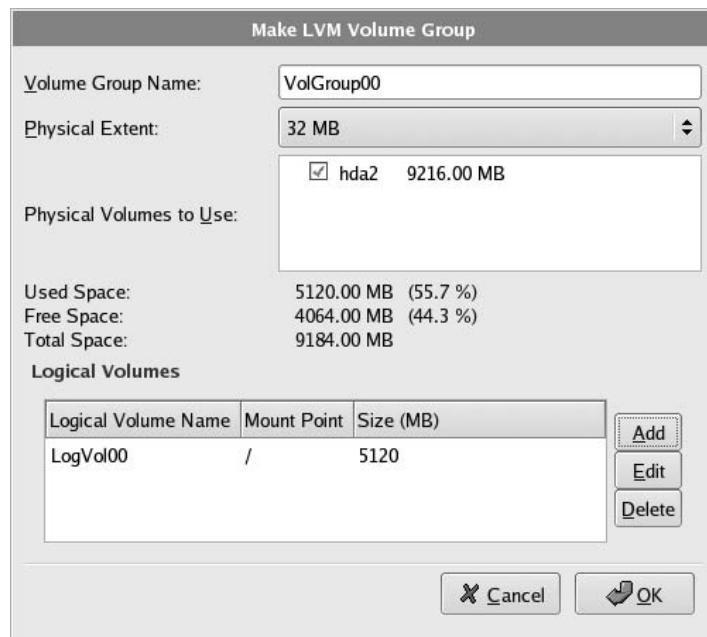
El cuadro de diálogo completado debe semejarse al que se muestra enseguida. Haga clic en el botón OK una vez que haya terminado.



6. Haga clic en el botón LVM. Aparecerá el cuadro de diálogo Make LVM Volume Group (Hacer grupo de volúmenes LVM). Acepte los valores predeterminados ya proporcionados para los diversos campos (Volume Group Name [Nombre del grupo de volúmenes], Physical Extent [Extensión física], etc.). Haga clic en Add. Aparecerá el cuadro de diálogo Make Logical Volume (Hacer volumen lógico). Complete los campos en el cuadro de diálogo con la información que sigue:

Mount Point	/
File System Type	ext3
Logical Volume Name	LogVol00
Size (MB)	5120 (aproximadamente 5GB)

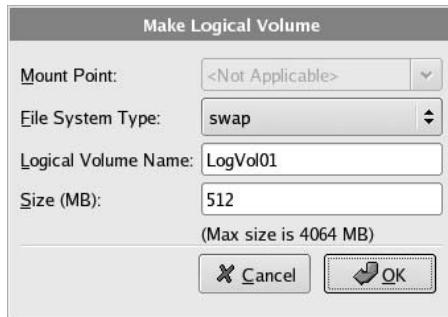
El cuadro de diálogo completado debe semejarse al que se muestra enseguida. Haga clic en el botón OK una vez que haya terminado.



7. Haga clic una vez más en Add, en el cuadro de diálogo Make LVM Volume Group. Aplicará el cuadro de diálogo Make Logical Volume. Complete los campos en el cuadro de diálogo con la información que sigue:

Mount Point	Deje en blanco
File System Type	swap
Logical Volume Name	LogVol01
Size (MB)	512 (aproximadamente el doble del total de la cantidad total de RAM disponible)

El cuadro de diálogo completado debe semejarse al que se muestra enseguida. Haga clic en el botón OK una vez que haya terminado.



8. Haga clic una vez más en Add en el cuadro de diálogo Make LVM Volume Group. Aparecerá el cuadro de diálogo Make Logical Volume. Complete los campos en el cuadro de diálogo con la información que sigue:

Mount Point	/home
File System Type	ext3
Logical Volume Name	LogVol02
Size (MB)	3072 (aproximadamente 3GB)

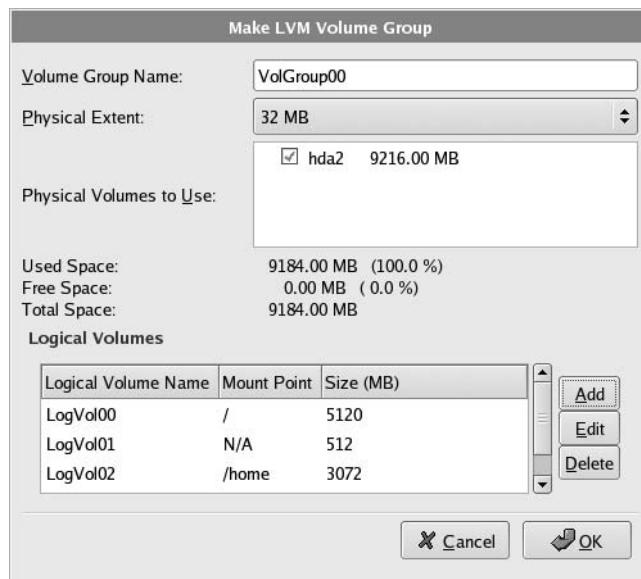
Haga clic en el botón OK una vez que haya terminado.

9. Haga clic una vez más en Add en el cuadro de diálogo Make LVM Volume Group. Aparecerá el cuadro de diálogo Make Logical Volume. Complete los campos en el cuadro de diálogo con la información que sigue:

Mount point	/tmp
File System Type	ext3
Logical Volume Name	LogVol03
Size (MB)	480 (o "Use up all the remaining free space on the Volume group")

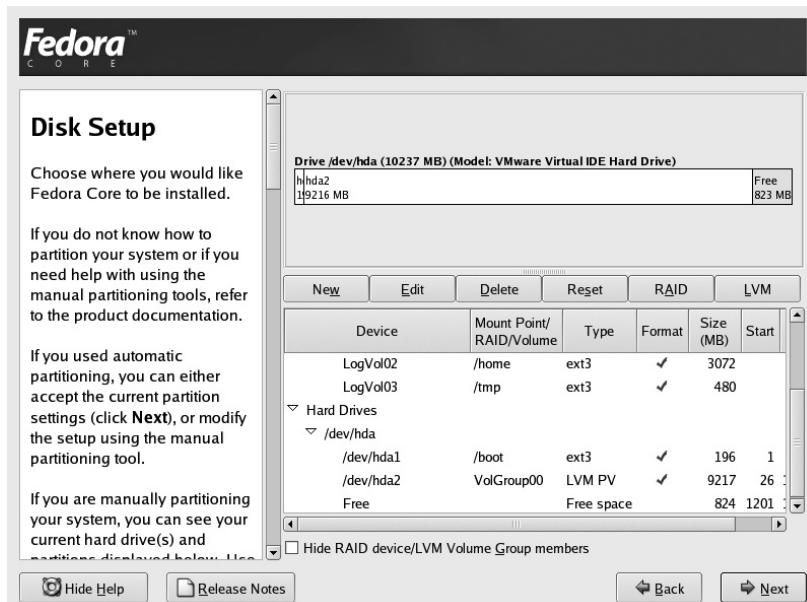
Haga clic en el botón OK una vez que haya terminado.

10. El cuadro de diálogo Make LVM Volume Group completado debe semejarse al que se muestra enseguida:



Haga clic en el botón OK para cerrar el cuadro de diálogo.

11. Se le regresará a la pantalla principal Disk Setup. La pantalla final debe ser semejante a la que se muestra enseguida:



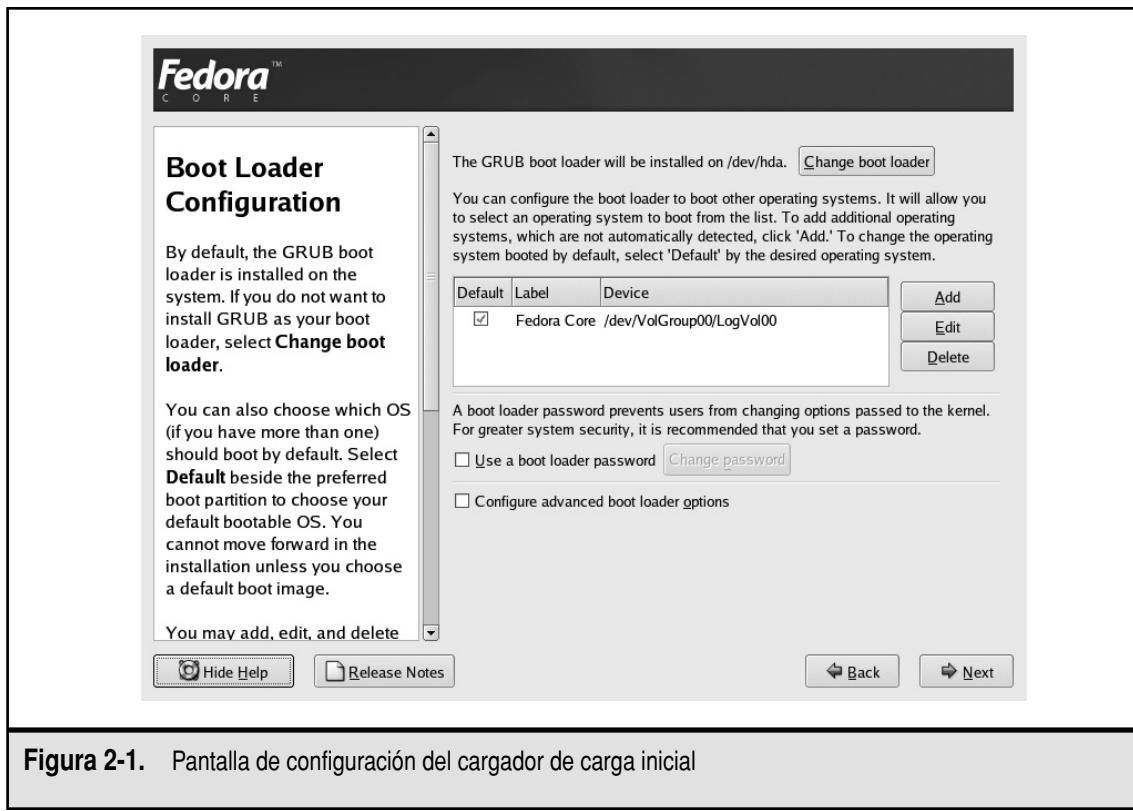
Advertirá que hemos dejado algo de espacio libre (*free*), sin particiones, debajo de la columna de dispositivo (Device). Esto se hizo de manera deliberada, de modo que el lector pueda jugar con ese espacio en los capítulos posteriores, sin tener que reinstalar todo el sistema operativo para crear el espacio libre.

12. Haga Clic en Next para completar la parte de partición del disco de la instalación.

Configuración del cargador de la carga inicial (boot)

GRUB es uno de los *administradores de la carga inicial* populares para Linux. Un administrador de la carga inicial maneja el proceso de arrancar en realidad el proceso de carga de un sistema operativo. Si están familiarizados con Windows NT, ya han tratado con el NT Loader (Cargador NT) (NTLDR), el cual presenta el menú en el momento de la carga inicial.

La pantalla de configuración del cargador de la carga inicial tiene múltiples secciones (vea la figura 2-1). En la parte superior de la pantalla se le dice dónde se está instalando el cargador de



carga inicial. En nuestro sistema de ejemplo, se está instalando en el Master Boot Record (Registro maestro de carga inicial) (MBR) de /dev/hda. El MBR es la verdadera primera instrucción que el sistema leerá cuando se lleva a cabo la inicialización de un sistema. En esencia, es el punto en donde finalizan las pruebas integradas del hardware y pasan el control al software.

También existe un botón en la parte superior de la pantalla que le permite cambiar el cargador de carga inicial. Lo normal es que, a menos que usted sepa en realidad lo que está haciendo, desee aceptar las opciones predeterminadas que se dan aquí.

La sección siguiente de la pantalla le permite configurar el cargador de carga inicial para inicializar otros sistemas operativos.

Si está instalando Linux en un disco duro que ya tiene algún otro sistema operativo (por ejemplo, Windows o alguna otra orientación de Linux), éste es en donde se configurará la funcionalidad de la inicialización dual. En un sistema que se configura para soportar tanto Windows como Linux, aquí verá sus elecciones. Si su sistema se estructura sólo para Linux (como se supone en este texto), verá una entrada.

NOTA La excepción es para los sistemas basados en SMP en el que verá dos entradas. La primera suele ser para soportar un sistema de múltiples procesadores, por lo común seguida por una entrada de respaldo para un núcleo con un solo procesador habilitado.

La última opción, rotulada como Configure Advanced Boot Loader Options (Opciones de configuración avanzada del cargador de la carga inicial), le permite introducir parámetros del núcleo que se van a usar en el momento de la carga inicial. La mayor parte de las personas pueden ignorar este cuadro. Si la documentación para una característica o dispositivo particulares requiere que pase aquí un parámetro, agréguelo; de lo contrario, deje sola la opción. Como reiteración, por lo común, la mayor parte de los valores predeterminados proporciona un buen trabajo para la mayor parte de las finalidades.

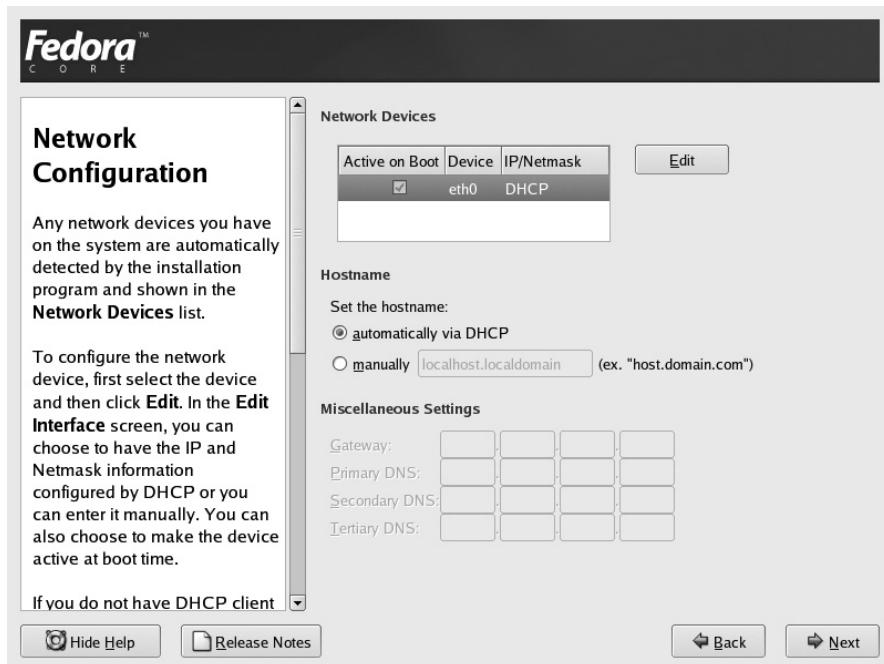
Acepte los valores predeterminados que se proporcionan y haga clic en Next.

Configuración de la red

Cada tarjeta de interfaz que se haya detectado en forma correcta aparecerá en la lista de la sección Network Devices (Dispositivos de la red). En Linux, los dispositivos de Ethernet se nombran eth0, eth1, eth2, y así sucesivamente. Para cada interfaz, puede configurarla usando DHCP o fijar en forma manual las direcciones IP. Si elige la configuración manual, asegúrese de tener listas las direcciones IP, de la netmask, de la red y de difusión.

En la mitad inferior de la pantalla, verá las selecciones de configuración para configurar el nombre del anfitrión del sistema, la interfaz del sistema e información DNS relacionada.

En nuestro sistema de ejemplo, vamos a configurar la primera interfaz de Ethernet –eth0– usando DHCP. Acepte todos los valores predeterminados en esta pantalla, como se muestra en la ilustración siguiente y haga clic en Next.



NOTA No se preocupe si sabe que no cuenta con un servidor DHCP disponible en su red con el que suministrará a su nuevo sistema la información de la configuración de IP. La interfaz de Ethernet sencillamente permanecerá no configurada. El nombre de anfitrión del sistema también se fijará de manera automática en localhost.localdomain, en ausencia de un servidor DHCP capaz que pueda asignar en forma automática los nombres de anfitriones.

Configuración de firewall (contrafuegos)

La rama siguiente de la instalación se refiere a las opciones de seguridad de las que se dispone.

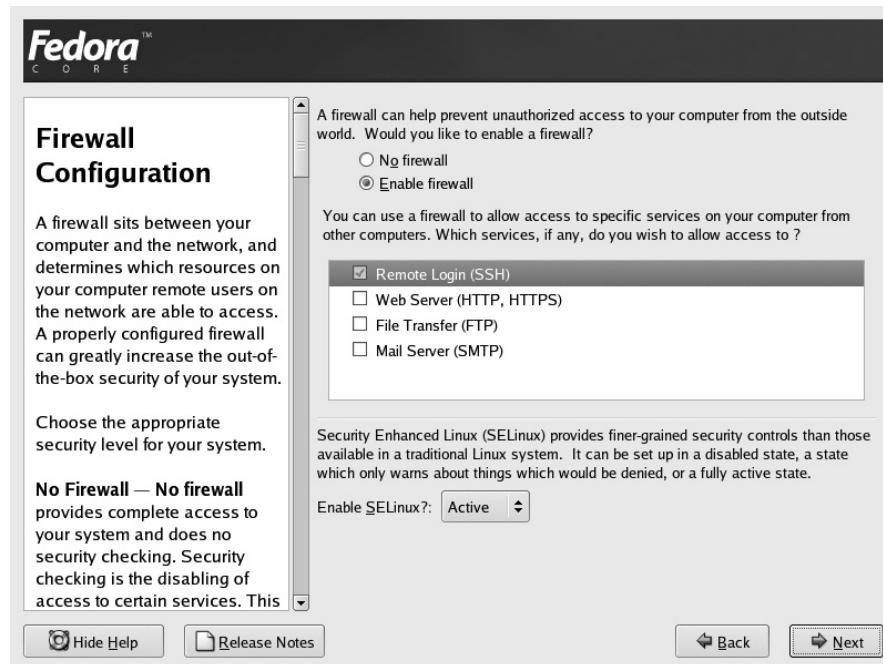
La primera opción le permite decidir si desea habilitar o no el firewall. Bajo esta sección también tiene la opción de configurar ese firewall para permitir el acceso a servicios específicos de los cuales su computadora es el anfitrión.

NOTA Esta sección de la instalación es una fuente común de confusión para los novatos en Linux. Los servicios de muestra [Remote Login (Concesión remota de acceso), Web Server, File Transfer (Transferencia de archivos), Mail Server (Servidor de correo)] se refieren a servicios que usted desea hacer funcionar en su sistema local y que tiene la intención de hacer accesibles desde el exterior (tráfico de ingreso). Por ejemplo, si quiere hacer funcionar un servidor Web en su servidor para ser anfitrión de un sitio Web, entonces puede abrir un agujero en el firewall para permitir el acceso al tráfico del servidor Web (HTTP, HTTPS). Dejar sin marca de verificación HTTP, HTTPS en esta pantalla *no* impedirá que su navegador Web presente las páginas Web cuyos anfitriones sean otros sistemas (tráfico de egreso).

La otra sección de la pantalla trata de la configuración de Linux mejorada respecto a la seguridad (SELinux).

SELinux proporciona controles de seguridad de grano más fino que aquellos de los cuales se dispone de manera tradicional. Durante la instalación, puede estructurar en un estado Disabled (Desactivado), un estado Active (Activado) o un estado Warn-Only (Sólo de alerta).

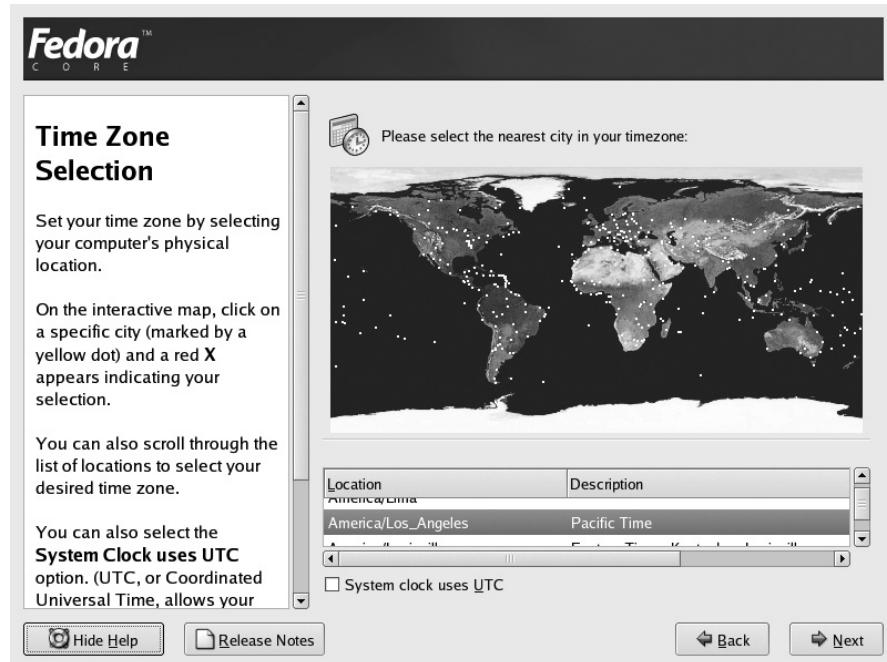
1. Acepte los valores predeterminados en esta pantalla pero, además, ponga la marca de verificación para permitir Remote Login (SSH) a través del firewall, como se muestra enseguida:



2. Haga clic en Next una vez que haya terminado.

Selección de la zona de tiempo

La pantalla de configuración de la zona de tiempo, mostrada a continuación, le permite seleccionar esa zona en la cual está ubicada la máquina:



Si el reloj del hardware de su sistema marca el tiempo en UTC, asegúrese de hacer clic en el cuadro de verificación System Clock Uses UTC (El reloj del sistema usa UTC) de modo que Linux puede determinar la diferencia entre los dos y presentar el tiempo local correcto.

1. Recorra la lista de lugares y seleccione la ciudad más cercana a su zona de tiempo. También puede usar el mapa interactivo para seleccionar una ciudad específica (marcada con un punto amarillo) para fijar su zona de tiempo.
2. Haga clic en Next una vez que haya terminado.

Fijación de la contraseña raíz

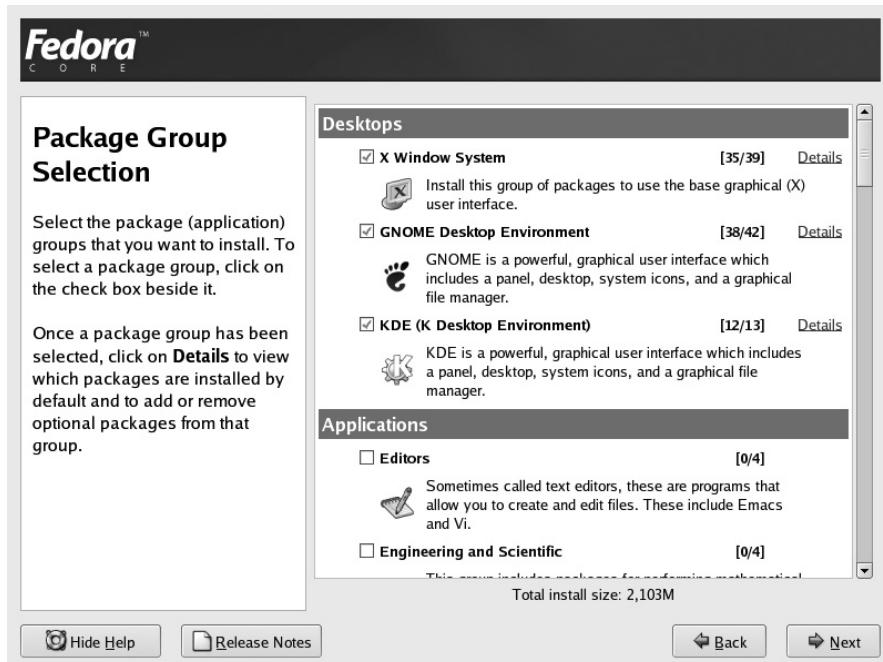
La parte siguiente de la instalación le permite fijar una contraseña para el usuario raíz, también conocido como superusuario. Es la cuenta más privilegiada del sistema y lo normal es que tenga el pleno control de éste. Es equivalente a la cuenta Administrator (Administrador) en el sistema operativo de Windows. Por tanto, resulta crucial que proteja esta cuenta con una buena contraseña. Asegúrese de no tomar palabras del diccionario o nombres como contraseñas, ya que son fáciles de adivinar y descifrar.

1. Tome una contraseña muy buena e intodúzcalo en el cuadro de texto Root Password (Contraseña raíz).
2. Introduzca una vez más la misma contraseña en el cuadro de texto Confirm (Confirmación).
3. Haga clic en Next.

Selección del grupo de paquetes

Ésta es la parte de la instalación en donde usted puede seleccionar cuáles paquetes (aplicaciones) va a instalar en el sistema. Fedora clasifica estos paquetes en varias categorías de alto nivel. Las categorías de las que se dispone son Desktops (Escritorios), Applications (Aplicaciones), Servers (Servidores), Development (Desarrollo), System (Sistema) y Miscellaneous (Diversos). Bajo cada categoría se encuentran los grupos de paquetes disponibles [X Window System, Editors (Editores), Development Tools (Herramientas de desarrollo), etc.]. Esta organización le permite hacer una selección rápida de cuál tipo de paquetes desea que estén instalados y, de modo seguro, ignora los detalles.

Mirando las selecciones que muestran aquí, puede ver el menú de los grupos de paquetes de alto nivel que Fedora le da. Sencillamente puede seleccionar los grupos que le interesan o trasladarse hasta la parte inferior de la lista y seleccionar Everything (Todo) para tener instalados todos los paquetes (una advertencia: ¡una instalación plena/todo puede requerir hasta 7GB de espacio de disco!), o bien, seleccionar la opción de instalar sólo el mínimo requerido.



El GNOME Desktop Environment (Entorno GNOME de escritorio) podría ya estar seleccionado para usted; GNOME es un entorno de escritorio muy popular. Además de los grupos de paquetes que se seleccionan en forma predeterminada, seleccione el grupo de paquetes KDE (K Desktop Environment). Esta selección adicional le permitirá muestrear algunos de los entornos disponibles de escritorio (interfaces gráficas del usuario, o GUI) de las que se dispone en Linux. Existe una antigua guerra santa referente a cuál de los ambientes de escritorio es el mejor, pero usted tendrá que ir jugando con ellos por allí para decidir por sí mismo.

1. Acepte los valores predeterminados y seleccione también el grupo de paquetes KDE (K Desktop Environment).
2. Haga clic en Next.

Acerca de la instalación

El instalador empezará la instalación real (disponiendo las particiones, formateando las particiones con un sistema de archivos, escribiendo el sistema operativo en el disco, etc.) después de que haga clic en Next en esta pantalla. Si se le ponen los pies fríos en este punto, todavía puede retirarse con seguridad de la instalación sin pérdida de datos (o de autoestima). Para salir del instalador, sencillamente restablezca su sistema presionando CTRL-ALT-DELETE en el teclado o presionando el interruptor de reposición o de energía eléctrica para el sistema.

1. Haga clic en Next para empezar.

NOTA Si está instalando a partir de un juego de CD, el instalador le informará de los discos en particular que necesita tener a la mano para completar la instalación. No recibirá esta advertencia si está llevando a efecto una instalación basada en la red o usando un DVD (los pasos que se están siguiendo aquí son los que se dan cuando se usa un DVD).

2. La instalación empezará y el instalador le mostrará el progreso de la misma, como se muestra a continuación:



Éste también es un buen momento para estudiar Release Notes (Emitir notas). Haga clic en el botón Release Notes en el lado izquierdo de la pantalla para lanzar otra ventana con el fin de ver las notas.

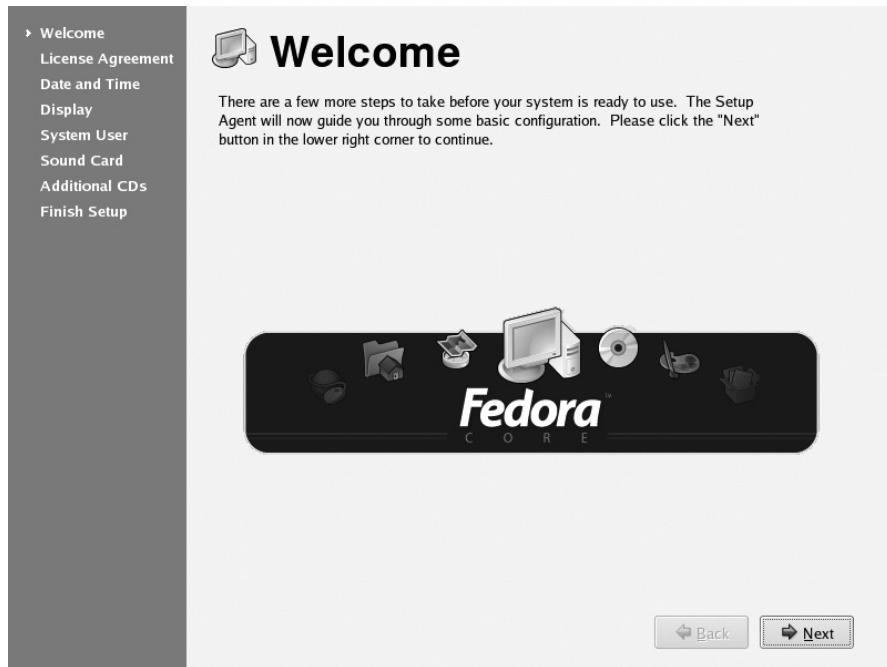
3. Elimine el medio de instalación de la unidad y haga clic en el botón Reboot (Reiniciar) en la pantalla final que se muestra enseguida:



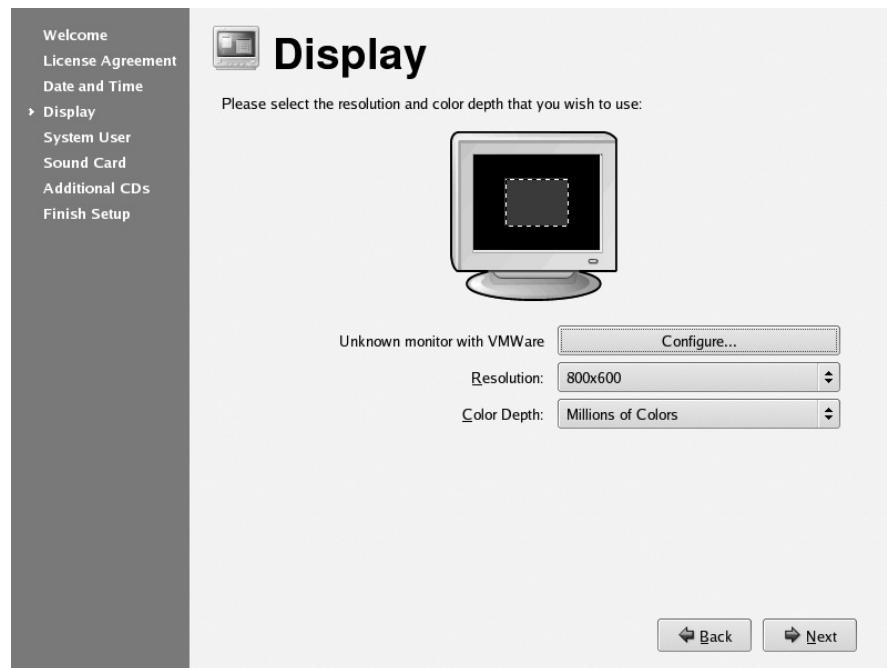
Configuración inicial del sistema

El sistema se reiniciará. Después de que se completa el proceso de reinicio, tendrá que hacer clic a través de un proceso de personalización [En Fedora y RHEL se le llama a este proceso “The Setup Agent” (El agente de estructuración)]. Éste es un proceso que se realiza una vez y no tendrá que pasar por él de nuevo en reinicios futuros. Aquí es donde usted puede fijar la hora y la fecha del sistema, agregar usuarios al mismo, instalar software adicional, y así sucesivamente.

1. Haga clic en Next cuando se presente con la pantalla Welcome (Bienvenido), mostrada a continuación:

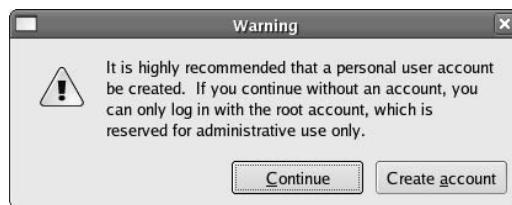


2. A continuación se le pedirá que lea y acepte el acuerdo de licencia, antes de seguir. Seleccione el botón de acción excluyente Yes, I Agree To The License Agreement (Sí, acepto el acuerdo de licencia) y haga clic en Next.
3. En la pantalla Date and Time (Fecha y hora) asegúrese que la fecha y hora actuales que se muestran reflejan las reales en el momento. Haga clic en Next cuando haya terminado.
4. En la sección Display (Presentación), que se muestra a continuación, el Setup Agent tratará de sondar el tipo de tarjeta de video y de monitor que ha puesto en el sistema. Si los valores detectados son incorrectos, puede usar los botones con el fin de seleccionar los correctos para su hardware. Si no está seguro de cualquier ajuste en particular, puede aceptar con seguridad los valores predeterminados que se sondaron automáticamente. Haga clic en Next cuando haya terminado.



5. A continuación, se le presentará un cuadro de diálogo con el cual puede crear una cuenta de usuario pero, en esta ocasión, pasará por alto la creación de algún usuario adicional. Haga clic en Next.

Aparecerá un cuadro de diálogo de advertencia, como se muestra abajo, en el que se le pide que cree una cuenta de usuario no privilegiado; seleccione Continue.



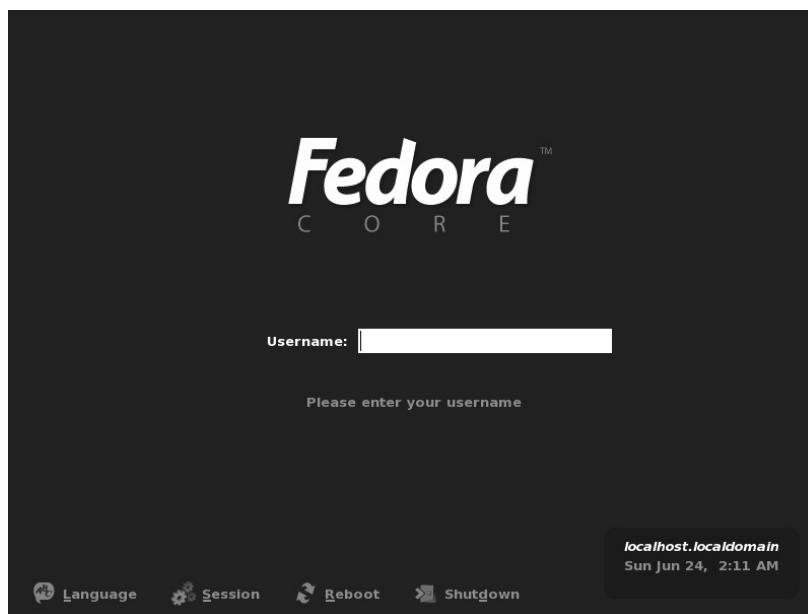
6. Si se detecta una tarjeta de sonido en su sistema, se le presentará la pantalla Sound Card. Haga clic en Play Test Sound (Realizar un sonido de prueba) si, en efecto, tiene una tarjeta de sonido en su sistema. Siga las instrucciones subsiguientes y haga clic en Next cuando haya terminado.
7. La sección Additional CDs (CD adicionales) le permite instalar paquetes adicionales que no instaló en el transcurso de la instalación y, en este punto, puede instalar algún software de tercera parte.

No tiene software adicional que quiera instalar, de modo que haga clic en Next.

8. Ahora el sistema está estructurado y listo para su uso. Haga clic en Next para salir del Setup Agent, mostrado abajo, y continúe



9. Se le presentará la pantalla de solicitud de acceso (login) de Fedora, mostrada a continuación:



RESUMEN

Ha completado con éxito el proceso de instalación. Si todavía tiene problemas con la instalación, asegúrese de visitar el sitio Web de Fedora en <http://fedora.redhat.com> y eche una mirada a los diversos manuales disponibles.

Las notas de emisión de la versión también constituyen un recurso muy bueno para la obtención de temas específicos relativos a la instalación. Aun cuando en el proceso de instalación que se discutió en este capítulo se usó Fedora Core como el sistema operativo elegido, puede descansar seguro de que los pasos de instalación para RHEL versión 4 son casi idénticos. Los pasos de la instalación también le presentaron algunos conceptos específicos de Linux/UNIX que se cubrirán con más detalle en capítulos posteriores (por ejemplo, convenciones de nombramiento del disco duro y particiones bajo Linux).

CAPÍTULO 3



Instalación
de software

El trabajo de los administradores de sistemas es la administración del software o de las aplicaciones en sistemas, de diversas maneras. Uno es de la clase de administradores de sistemas que les gusta desempeñarse dentro de lo seguro y, en general, se atienen al principio de “si no está descompuesto, no hay que arreglarlo”. Este modo de proceder tiene sus beneficios así como sus desventajas. Uno de los beneficios es que el sistema tiende a ser más estable y se comporta de una manera que se puede predecir. Nada ha cambiado de modo drástico en el sistema y, por tanto, debe estar en gran parte de la misma manera en que estuvo ayer, la semana pasada, el mes pasado, etcétera. La desventaja de este modo de pensar es que el sistema podría perder los beneficios de la eliminación de errores y de mejora de la seguridad de los que se dispone para las diversas aplicaciones instaladas.

Otra clase de administradores de sistemas toman el enfoque diametralmente opuesto: les gusta instalar la pieza más reciente y más grande de software que exista fuera de allí. Esta forma de ver las cosas también tiene sus beneficios y sus desventajas. Uno de sus beneficios es que el sistema tiende a estar al corriente conforme se descubren y arreglan las fallas relativas a la seguridad en las aplicaciones. La desventaja obvia es que algo del software más reciente podría no haber tenido tiempo de recibir los beneficios del proceso de maduración que viene con la edad y, por lo cual, puede comportarse de maneras un poco imprevisibles.

Sin importar cuál sea su estilo de administración del sistema, hallará que gran parte de su tiempo se consumirá interactuando con los diversos componentes de software del sistema, manteniéndolos actualizados, dando mantenimiento a los que ya ha instalado o agregando software nuevo.

Hay un par de procedimientos básicos para instalar software en un sistema Linux. Uno de ellos es usar el administrador de paquetes para la distribución. Un método común para Fedora Core y Red Hat Enterprise Linux es usar el Red Hat Package Manager (Administrador de paquetes de Red Hat) (RPM). Otro procedimiento es compilar e instalar el software a mano, usando el método estándar de compilación GNU, o bien, las directivas específicas del software. En este capítulo, cubriremos estos métodos.

EL RED HAT PACKAGE MANAGER

El Red Hat Package Manager (RPM) permite la instalación y remoción fáciles de paquetes de software (por lo general, software compilado previamente). Un paquete consta de un archivo de archivos y otros metadatos.

Es maravillosamente fácil de usar y, en torno a él, se han construido varias interfaces gráficas para hacerlo todavía más fácil. En varias distribuciones de Linux y en varias terceras partes se usa esta herramienta para distribuir y empaquetar su software. De hecho, casi todo el software mencionado en este libro se encuentra en forma RPM. La razón por la que pasará por el proceso de compilar software por sí mismo en otros capítulos es para que usted pueda personalizar el software para su sistema, ya que podría ser que esas personalizaciones no se lograran con facilidad en un RPM.

Un archivo RPM es un paquete que contiene los archivos necesarios para que el software funcione en forma correcta. Estos archivos pueden ser de configuración, binarios e incluso scripts previos y posteriores para ejecutarse en el transcurso de la instalación del software.

NOTA En el contexto presente, estamos suponiendo que los archivos RPM contienen binarios precompilados, pero, adhiriéndose al principio de fuente abierta, las diversas distribuciones comerciales de Linux están obligadas a hacer que se encuentre disponible el código fuente para la mayor parte de los binarios GNU (algunos vendedores de Linux se apegan a este principio más que otros). Por lo tanto, varios vendedores de Linux hacen que el código fuente se encuentre disponible para sus binarios, en la forma RPM. Por ejemplo, Fedora y SuSE ponen a disponibilidad el código fuente como un RPM, y se está volviendo cada vez más común descargar y compilar el código fuente en esta forma.

La herramienta RPM realiza la instalación y desinstalación de programas RPM. La herramienta también mantiene una base central de datos de aquellos RPM que haya instalado, en dónde están instalados y otra información acerca del paquete.

En general, el software que viene en la forma de un RPM da menos trabajo para ser instalado y para mantenerlo que aquel que necesita ser compilado. La compensación es que al usar un RPM, acepta los parámetros predeterminados que se suministran en el RPM. En la mayor parte de los casos, estos valores predeterminados son aceptables. Sin embargo, si necesita tener un conocimiento más íntimo de lo que está pasando con una parte del software, puede hallar que al compilar por sí mismo la fuente, aprenderá más acerca de qué componentes y opciones del paquete existen y cómo funcionan juntos.

Suponiendo que todo lo que usted quiere es un paquete sencillo, RPM es perfecto. Existen varios grandes recursos para los paquetes RPM, incluyendo los siguientes:

- ▼ <http://www.rpmfind.net>
- <http://rpm.pbone.net>
- <ftp://ftp.redhat.com>
- <http://mirrors.kernel.org>
- ▲ <http://freshrpms.net>

Por supuesto, si tiene interés en obtener más detalles acerca del propio RPM, puede visitar el sitio Web de RPM en <http://www.rpm.org>. RPM viene con Fedora Core, SuSE, Mandrake, otros incontables derivados de Red Hat y, lo más sorprendente de todo, ¡con la versión de Red Hat de Linux! Si no está seguro de si viene RPM con su distribución, verifíquelo con su vendedor.

NOTA Aunque el nombre del paquete diga "Red Hat", el software también puede ser utilizado con otras distribuciones. De hecho, RPM incluso ha sido acoplado a otros sistemas operativos, como Solaris e IRIX. El código fuente para RPM es software de fuente abierta, de modo que quienquiera puede tomar la iniciativa de hacer que el sistema trabaje para ellos.

Las funciones primarias del Red Hat Package Manager son

- ▼ Software de consulta, instalación y desinstalación
- Mantenimiento de una base de datos que almacena varios artículos de información acerca del paquete
- ▲ Empaque de otro software en una forma RPM

En las secciones siguientes, se examinan algunas de estas funciones mediante el uso de ejemplos reales.

La tabla 3-1, la cual incluye las opciones RPM usadas con frecuencia, se suministra sólo para fines de consulta.

Opción de la línea de comandos	Descripción
--install	Ésta instala un nuevo paquete.
--upgrade	Ésta actualiza el paquete instalado a una versión más reciente o instala esta última.
--erase	Elimina o borra un paquete instalado.
--query	Ésta es la opción usada para consulta.
--force	Ésta es el marro de la instalación. Por lo general, la usa cuando está instalando con conocimiento una configuración rara o desacostumbrada y los salvaguardas de RPM están intentando evitar que lo haga. La opción --force le dice a RPM que se abstenga de realizar cualesquiera pruebas de sanidad y sólo que lo haga, incluso si piensa que usted está intentando meter una clavija cuadrada en un hoyo redondo. Tenga cuidado con esta opción.
-h	Imprime marcas semejantes a galones para indicar el progreso en el transcurso de una instalación. Utilícela con la opción -v para tener una presentación bonita.
--percent	Imprime el porcentaje completado para indicar el progreso. Resulta práctica si está ejecutando RPM desde otro programa, como un script Perl y desea saber el estado de la instalación.
-nodeps	Si RPM se está quejando acerca de archivos de dependencia faltantes pero usted quiere que, de cualquier manera, se lleve a efecto la instalación, el paso de esta opción a la línea de comandos hará que RPM no realice comprobaciones de dependencia.
-q	Pide información al sistema RPM.
--test	Esta opción no realiza una instalación real; sólo hace la comprobación para ver si tendría éxito una instalación. Si anticipa problemas, presenta cuáles serán.
-v	Verifica los RPM o archivos en el sistema.
-v	Le dice a RPM que exprese en forma muy amplia sus acciones.

Tabla 3-1. Opciones comunes de RPM

Consulta con RPM (empezar a conocerse entre sí)

Una de las mejores maneras de empezar cualquier relación es conocer a la otra parte. Algo de la información pertinente podría ser el nombre de la persona, qué hace esa persona para vivir, la fecha de nacimiento, lo que le gusta o le disgusta, etcétera. Las mismas reglas se aplican a los paquetes del tipo RPM. Después de que obtiene una pieza del software (de Internet, del CD/DVD de la distribución, de una tercera parte, etc.), debe conocer el software antes de hacer que forme parte de su vida...perdón, de su sistema. Esta funcionalidad se construyó en RPM desde el principio, y es muy fácil de usar.

Cuando haya usado Linux/UNIX, puede ser que encuentre que los nombres del software son un tanto intuitivos y que, por lo general, puede decir lo que es un paquete sólo con mirar su nombre. Por ejemplo, puede ser que al no iniciado no le sea obvio de inmediato que un archivo nombrado **gcc-3.8.2-6.rpm** sea un paquete para la "colección del compilador GNU". Pero una vez que haya usado el sistema y sepa lo que busca, se vuelve más intuitivo. También puede usar RPM para consultar otros tipos de información, como la fecha de estructuración del paquete, su peso...perdón, su tamaño, lo que le gusta o le disgusta...perdón, sus dependencias, etcétera.

Empecemos por intentar unas cuantas cosas usando RPM. Iniciemos con el ingreso al sistema y el arranque de una terminal (como se describe en el texto siguiente: "Ponerse a trabajar en el negocio").

Ponerse a trabajar en el negocio

En el capítulo anterior se le hizo caminar por el proceso de instalación. Ahora que tiene un sistema para trabajar, necesitará ingresar al sistema para llevar a cabo los ejercicios de este capítulo y de otros capítulos del libro. La mayor parte de los ejercicios le pedirá implícitamente que teclee un comando. Aun cuando parece como hablar de lo obvio, siempre que se le pida teclear un comando, tendrá que teclearlo dentro de una consola en el mensaje del shell (intérprete del lenguaje de comandos). Éste es semejante al mensaje DOS en Windows de Microsoft, pero estupendamente más poderoso.

Existen un par de maneras de hacer esto. Una de ellas es usar una terminal en una bonita ventana (GUI), o la consola del sistema. Las consolas en ventanas se conocen como emuladores de terminales (pseudoterminales) y existen toneladas de ellas.

Después de ingresar a su escritorio elegido (GNOME, KDE, XFCE, etc.), por lo común puede lanzar una pseudoterminal al hacer clic derecho sobre el escritorio y seleccionar Launch Terminal (Lanzar terminal) a partir del menú sensible al contexto. Si no tiene esa opción particular, busque una opción en el menú que dice Run Command (Ejecutar comando). Después de que aparece el cuadro de diálogo Run, entonces puede teclear el nombre de un emulador de terminal en el cuadro de texto Run. Un programa popular emulador de terminales que es casi garantizado (yo se le reembolsa su dinero!) que exista en todos los sistemas de Linux es el venerable **xterm**. Si usted se encuentra en un escritorio GNOME, el **gnome-terminal** es el valor predeterminado. Si está usando KDE, el valor determinado es **konsole**.

Consulta respecto a todos los paquetes

Use el comando **rpm** para obtener una lista de todos los paquetes que están instalados actualmente en su sistema. En el mensaje del shell, teclee

```
[root@serverA ~]# rpm --query -all
```

Esto le dará una larga lista del software instalado.

NOTA Como la mayor parte de los comandos de Linux, el comando **rpm** también tiene sus formas largas o cortas (abreviadas) de las opciones o argumentos. Por ejemplo, la forma corta de la opción **--query** es **-q**, y la forma corta de **-all** es **-a**. La mayor parte de las veces, en este libro usaremos las formas cortas pero, en ocasiones, usaremos las largas sólo de modo que usted pueda ver su relación.

Consulta de detalles de un paquete específico

Sea cero en uno de los paquetes cuya lista obtuvimos en la salida del comando precedente, la aplicación bash. Use **rpm** para ver si, en efecto, tiene la aplicación bash instalada en su sistema.

```
[root@serverA ~]# rpm --query bash  
bash-3.0-17
```

La salida debe ser algo semejante a esto. Muestra que, en efecto, tiene instalado el paquete llamado bash. También muestra el número de versión anexado al nombre del paquete. Note que el número de versión de la salida en su sistema podría ser diferente. Pero el nombre del paquete principal casi siempre será el mismo, es decir, bash es bash es bash es bash en SuSE, Fedora, Mandrake, RHEL, etcétera.

Lo cual nos lleva a la pregunta siguiente: ¿qué es bash y qué hace? Para averiguarlo, teclee

```
[root@serverA ~]# rpm -qi bash  
Nombre      : bash          Reubicaciones: /usr  
Versión     : 3.0           Vendedor: Red Hat, Inc.  
Emisión     : 17            Fecha de la estructuración: Jueves 19 de octubre de 2004 12:40 PM PDT  
Fecha de  
instalación : Sábado 22 de enero de 2007 08:15:12 PM PST  
Estructuración del  
anfitrión   : bugs.build.redhat.com  
Grupo       : System Environment/Shells    Fuente RPM: bash-3.0-17.src.rpm  
Tamaño      : 5112768        Licencia: GPL  
Firma       : DSA/SHA1, Miércoles 20 de octubre de 2005 09:07:11 AM PDT, Clave ID b44269d04f2a6fd2  
Empaquetador: Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>  
Resumen     : El Bourne Again shell de GNU (bash).  
Descripción :  
El proyecto Bourne Again shell (bash) de GNU es un shell o intérprete del  
lenguaje de comandos que es compatible con el programa shell (sh) de Bourne.  
Bash incorpora características útiles del shell de Korn (ksh) y del shell C (csh)  
y bash puede ejecutar la mayor parte de los scripts (guiones) sin modificación.  
Bash es el shell predeterminado para Linux de Red Hat.)
```

Esta salida nos da una gran cantidad de información. Muestra el número de versión, la emisión, la descripción, el empaquetador y más.

El paquete bash parece un tanto impresionante. Veamos qué más viene con él.

```
[root@serverA ~]# rpm -ql bash
```

Con esto se produce una lista de todos los archivos que viene con el paquete bash.

Para obtener una lista de los archivos de configuración (si los hay) que viene con el paquete bash, teclee

```
[root@serverA ~]# rpm -qc bash
/etc/skel/.bash_logout
/etc/skel/.bash_profile
/etc/skel/.bashrc
```

Las capacidades de consulta de **rpm** son muy extensas. Los paquetes RPM tienen almacenada una gran cantidad de información en las llamadas TAG. Estas etiquetas (*tags*) integran los metadatos del paquete. Usted puede consultar la base de datos del RPM, en busca de información específica, usando estas etiquetas. Por ejemplo, para averiguar la fecha en que el paquete bash se instaló en su sistema, puede teclear

```
[root@serverA ~]# rpm -q --qf "[ %{INSTALLTIME:date} \n]" bash
```

Sábado 22 de enero de 2007 08:15:12 PM PST



NOTA Debido a que bash es una parte estándar de las distribuciones de Linux y se habrían instalado cuando usted instaló inicialmente el OS, encontrará que esta fecha de instalación estará muy cercana al día en el que instaló ese OS.

Para averiguar bajo cuál grupo de paquetes viene la aplicación bash, teclee

```
[root@serverA ~]# rpm -q --qf "[ %{GROUP} \n]" bash
```

System Environment/Shells

Por supuesto, siempre puede consultar acerca de más de un paquete al mismo tiempo y también consultar acerca de información de múltiples etiquetas. Por ejemplo, para presentar los nombres y grupos de paquetes para los paquetes bash y xterm, teclee

```
[root@serverA ~]# rpm -q --qf "[ %{NAME} - %{GROUP} - %{SUMMARY} \n]" bash xterm
bash - System Environment/Shells - El Bourne Again shell (bash) de GNU.
xterm - User Interface/X - emulador xterm de terminales para el X Window System
```

Con el fin de determinar cuáles otros paquetes del sistema dependen del paquete bash, teclee

```
[root@serverA ~]# rpm -q --whatrequires bash
```

SUGERENCIA Las consultas RPM que se acaban de hacer se llevaron a cabo en el software que está instalado actualmente en el sistema. Puede realizar también consultas semejantes en otro software que obtenga de otras fuentes, por ejemplo, software que esté planeando instalar y que haya obtenido de Internet o del CD/DVD de la distribución. También se pueden efectuar consultas semejantes en paquetes que todavía no se hayan instalado. Para hacer esto, sencillamente agregue la opción **-qP** al final del comando de consulta. Por ejemplo, para hacer una consulta a un paquete no instalado cuyo nombre sea “*joe-3.1-8.i386.rpm*,” teclearía **rpm -qip joe-3.1-8.i386.rpm**.

Instalación con RPM (frecuentarse)

Muy bien, ahora los dos están listos para llevar la relación hacia la etapa siguiente. Los dos han decidido frecuentarse. Esto puede ser algo bueno porque les permite a los dos ver y probar qué tan compatibles son en verdad.

Esta etapa de las relaciones se parece a instalar el paquete de software en su sistema; es decir, mover el software hacia su sistema.

En los pasos siguientes del proyecto, instalará la aplicación llamada “*joe*” en su sistema. En primer lugar, necesitará obtener una copia del paquete RPM para *joe*. Puede obtener este programa de varios lugares (los CD/DVD de instalación, de Internet, etcétera). En el ejemplo que sigue, se usa una copia del programa que vino con el DVD usado en el transcurso de la instalación.

Es necesario montar el CD/DVD para tener acceso a su contenido. Para montarlo, inserte el DVD en la unidad y lance una consola. Si no ve que aparezca un ícono para el DVD en el escritorio después de un lapso breve, entonces tendrá que montar en forma manual el dispositivo. En el mensaje del shell, teclee

```
[root@serverA RPMS]# mount /media/dvd
```

o bien,

```
[root@serverA RPMS]# mount /media/cdrom
```

Los archivos RPM se almacenan bajo el directorio **Fedora/RPMS**, bajo del punto de montaje de su dispositivo de DVD/CD; por ejemplo, el directorio **/media/dvd/Fedora/RPMS**.

NOTA Si no tiene un CD o DVD de Fedora Core, puede descargar el RPM que estaremos usando en la sección siguiente de <http://mirrors.kernel.org/fedora/core/4/i386/os/Fedora/RPMS/joe-3.1-8.i386.rpm>.

Pasemos por los pasos de instalación de un RPM.

1. Lance una terminal virtual.
2. Suponiendo que su disco de instalación de la distribución se monta en el punto de montaje **/media/dvd**, cambie al directorio que suele contener los paquetes RPM en el DVD, Teclee

```
[root@serverA ~]# cd /media/dvd/Fedora/RPMS/
```

En principio, asegúrese de que el archivo que quiere en realidad está en ese directorio. Use el comando **ls** para obtener una lista de todos los archivos que principian con las letras “*joe*” en el directorio. Teclee

```
[root@serverA RPMS]# ls joe*
joe-3.1-8.i386.rpm
```

NOTA Si está usando los CD de instalación, en lugar del DVD, el archivo podría estar ubicado en el segundo CD del juego de cuatro.

3. Ahora que ha confirmado que el archivo está allí, efectúe una prueba de instalación del paquete (esto se ejecutará recorriendo todos los movimientos de instalación del paquete, sin instalar en realidad algo en el sistema). Esto resulta útil para adquirir la seguridad de que se satisfacen todas las necesidades (dependencias) del paquete. Teclee

```
[root@serverA RPMS]# rpm --install --verbose --hash --test joe-3.1-8.i386.rpm
advertencia: joe-3.1-8.i386.rpm: V3 DSA firma: NOKEY, clave ID 4f2a6fd2
Preparación... #####
[100%]
```

4. Siga adelante y efectúe la instalación real. Teclee

```
[root@serverA RPMS]# rpm -ivh joe-3.1-8.i386.rpm
advertencia: joe-3.1-8.i386.rpm: V3 DSA firma: NOKEY, clave ID 4f2a6fd2
Preparación... #####
[100%]
1:joe #####
[100%]
```

5. Ejecute una consulta sencilla para confirmar que la aplicación está ahora instalada en su sistema. Teclee

```
[root@serverA RPMS]# rpm -q joe
joe-3.1-8
```

La salida hace ver que joe ahora está disponible en el sistema. Joe es un sencillo editor de textos. Puede lanzarlo sencillamente al teclear **joe** en el mensaje del shell. Para salir de joe, presione **CTRL-SHIFT-C**.

Como puede ver, la instalación de paquetes a través de RPM puede ser muy fácil. Pero se presentan veces en las que la instalación de paquetes es un poco más difícil. Esto suele deberse a los aspectos de dependencias que fallan. Por ejemplo, el paquete joe podría requerir que el paquete bash ya esté instalado en el sistema antes de que él mismo se pueda instalar con éxito.

Recorramos los pasos de instalación de un paquete más complejo para ver cómo se manejan las dependencias con RPM. Suponiendo que todavía se encuentre en el directorio Fedora RPMS, haga lo siguiente:

1. Instale el paquete al teclear

```
[root@serverA RPMS]# rpm -ivh gcc-4*.rpm
advertencia: gcc-4*.rpm: V3 DSA firma: NOKEY, clave ID 4f2a6fd2
error: Las dependencias fallaron:
       gcc-4* necesita glibc-devel >= 2.2.90-12
```

La salida precedente no se mira muy buena. En el último renglón se nos dice que **gcc*** depende de otro paquete, llamado **glibc-devel***.

2. Por fortuna, debido a que tenemos acceso al DVD que contiene todos los paquetes para esta distribución, en un solo directorio, podemos agregar con facilidad el paquete adicional a la lista de nuestra instalación. Teclee

```
[root@serverA RPMS]# rpm -ivh gcc-4*.rpm \
glibc-devel-*.rpm
warning: gcc-4*.rpm: V3 DSA signature: NOKEY, key ID 4f2a6fd2
error: Las dependencias fallaron:
        glibc-headers is needed by glibc-devel-*
        glibc-headers = 2.3.3 is needed by glibc-devel-*
```

Ah, ah... En apariencia este patrón particular no va a ser tan fácil de meter. En la salida se nos vuelve a decir que el paquete glibc-devel* depende de otro paquete, llamado glibc-headers*.

3. Agregue la dependencia más reciente a la lista de la instalación. Teclee

```
[root@serverA RPMS]# rpm -ivh gcc-4*.rpm \
glibc-devel-*.rpm glibc-headers-*.rpm
warning: gcc-4*.rpm: V3 DSA signature: NOKEY, key ID 4f2a6fd2
error: Las dependencias fallaron:
        kernel-headers is needed by glibc-headers-*
        kernel-headers >= 2.2.1 is needed by glibc-headers-*
```

Después de todo lo que le hemos dado a esta relación, todo lo que tenemos es más quejas. El último requisito es uno de aquellos nombres no intuitivos de paquetes con los que rara vez se cruzará, debido a que la dependencia en realidad se satisface por medio de un paquete llamado "glibc-kernheaders*" y *no* "kernel-headers, como podríamos haber pensado.

4. Parece que estamos cercanos al final. Agregamos el paquete requerido final a la lista. Teclee

```
[root@serverA RPMS]# rpm -ivh gcc-4*.rpm \
glibc-devel-*.rpm glibc-headers-*.rpm \
glibc-kernheaders-*.rpm
warning: gcc-4*.rpm: V3 DSA signature: NOKEY, key ID 4f2a6fd2
Preparing...
1:glibc-kernheaders ###### [100%]
2:glibc-headers ###### [ 25%]
3:glibc-devel ###### [ 50%]
4:gcc ###### [ 75%]
```

Fue duro, pero hizo lo necesario para tener instalado el software.

SUGERENCIA Cuando realiza instalaciones RPM múltiples de una sola vez, como lo hizo en el paso anterior, esto se conoce como una *transacción RPM*.

Una opción muy popular usada en la instalación de paquetes a través de RPM es la opción **-U** (por Upgrade, Actualizar). Es útil en especial cuando quiere instalar una versión más reciente de un paquete que ya existe. Sencillamente actualizará el paquete ya instalado a la versión más reciente. Esta opción también realiza un buen trabajo en conservar intacta su configuración personal para una aplicación. Por ejemplo, si hubiera tenido instalado joe-7-8.rpm y deseado actualizar a joe-8.0-rpm, teclearía **rpm -Uvh joe-8.0.rpm**.

Asimismo se debe hacer notar que también puede usar la opción **-U** para realizar una instalación normal de un paquete, incluso cuando no está actualizando.

Desinstalación con RPM (finalización de la relación)

Las cosas no funcionaron bien de la manera que ambos habían anticipado. Ahora es el momento de finalizar la relación. De cualquier manera, el otro cónyuge nunca sirvió para algo... de modo que sencillamente saquémoslos de nuestro sistema (literalmente).

Limpiar a fondo, a su manera, es una de las áreas en las que RPM en verdad sobresale, y éste es uno de sus puntos clave para su venta como administrador de software en los sistemas de Linux. Debido a que se almacena y mantiene una base de datos de varias piezas de información junto con cada paquete instalado, resulta fácil para RPM consultar su base de datos para reunir la información acerca de lo que se instaló y en dónde.

NOTA En este punto, cabe una pequeña advertencia. En forma muy semejante con las herramientas de instalación/desinstalación de Windows, todas las cosas maravillosas que RPM puede hacer también dependen del empaquetador del software. Por ejemplo, si la aplicación estuviera mal empacada y sus scripts de remoción no estuvieran formateados de manera apropiada, entonces todavía podría finalizar con trocitos del paquete en su sistema, incluso después de la desinstalación. Ésta es una de las razones por la que siempre debe adquirir su software de fuentes confiables.

La eliminación del software con RPM es bastante fácil y se puede hacer en un solo paso. Por ejemplo, para eliminar el paquete joe que se instaló con anterioridad, sencillamente necesitamos usar la opción **-e** como sigue:

```
[root@serverA RPMS]# rpm -e joe
```

Este comando no suele darle retroalimentación si todo fue bien. Para obtener más palabrería de salida para el proceso de desinstalación, agregue la opción **-vvv** al comando.

Una característica práctica de RPM es que le protegerá contra la eliminación de paquetes que necesitan otros paquetes. Por ejemplo, si tratamos de eliminar el paquete glibc-headers (algo del que vimos que gcc depende), veríamos lo siguiente:

```
[root@serverA RPMS]# rpm -e glibc-headers
error: Las dependencias fallaron:
        glibc-devel-* (instalado) necesita glibc-headers
        glibc-devel-* (instalado) necesita glibc-headers = 2.3.3
```

NOTA Recuerde que el paquete glibc-headers* requirió este paquete. Y, por consiguiente, RPM dará lo mejor de sí para ayudarle a mantener estable el entorno del software. Pero si usted es firme y está desesperado por pegarse un tiro, RPM también le permitirá hacerlo (quizá porque sabe que usted lo está haciendo). Si, por ejemplo, desea desinstalar a fuerzas el paquete glibc-devel, agregaría la opción **--nodeps** al comando de desinstalación.

Otras cosas que puede hacer con RPM

Además de la instalación y desinstalación básicas de paquetes con RPM, existen también otras numerosas cosas que puede hacer con él. En esta sección, haremos un recorrido sobre lo que son algunas de estas funciones.

Verificación de paquetes

Una opción muy útil para la herramienta RPM es la capacidad de verificar un paquete. Lo que sucede es que RPM mira la información del paquete en su base de datos, la cual se supone es buena. Entonces, compara esa información con los binarios y los archivos que se encuentran en el sistema de usted.

En el mundo actual de Internet, en donde algún intruso se meta a su sistema es una posibilidad real, esta clase de prueba debe decirle en forma instantánea si alguien le ha hecho algo a su sistema.

Por ejemplo, con el fin de verificar que el paquete bash está como debe estar, teclee

```
[root@serverA ~]# rpm -V bash
```

La ausencia de cualquier salida es un buen signo.

También puede verificar archivos específicos en el sistema de archivos que instaló un paquete en particular. Por ejemplo, para verificar si el comando **/bin/ls** es válido, teclearía

```
[root@serverA ~]# rpm -Vf /bin/ls
```

Una vez más, la falta de salida es una cosa buena.

Si algo estuviera fuera de lugar, por ejemplo, si el comando **/bin/ls** ha sido reemplazado por una versión falsificada, la salida de la verificación podría ser semejante a la que se da enseñada:

```
[root@serverA ~]# rpm -Vf /bin/ls
SM5....TC    /bin/ls
```

Si algo está mal, como en el ejemplo anterior, RPM le informará cuál prueba falló. Algunas pruebas de ejemplo son la prueba de suma de comprobación de MD5, el tamaño del archivo y los tiempos de modificación. La moraleja de la historia es: RPM es un aliado en la averiguación de lo que está mal en su sistema.

En la tabla 3-2, se da un resumen de los diversos códigos de error y su significado.

Si desea verificar todos los paquetes instalados en su sistema, teclee

```
[root@serverA ~]# rpm -Va
```

Con este comando se verifican *todos* los paquetes instalados en su sistema. Eso es una gran cantidad de archivos, de modo que podría tener que darle algo de tiempo para que se complete.

Código	Significado
S	Difiere el tamaño de archivo
M	Difiere el modo (incluye permisos y tipo de archivo)
5	Difiere la suma de MD5
D	No corresponden los números principal/menor de dispositivo
L	No corresponde la trayectoria de readLink
U	Difiere la propiedad del usuario
G	Difiere la propiedad del grupo
T	Difiere mTime

Tabla 3-2. Atributos de error de la verificación RPM

Validación de paquetes

Otra característica de RPM es que los paquetes se pueden firmar en forma digital. Esto proporciona una forma de mecanismo de autenticación integrado que permite a un usuario cerciorarse de que el paquete que se encuentra en su posesión en verdad fue empacado por la parte que él piensa que le proporcionó el paquete y también que no se intentó forzar este último en alguna parte a lo largo de la línea.

A veces, usted necesita decirle en forma manual a su sistema en cuál firma digital debe confiar. Esto explica las advertencias en los primeros proyectos cuando estaba tratando de instalar un paquete (como este mensaje: “warning: joe-3.1-8.i386.rpm: V3 DSA signature: NOKEY, key ID 4f2a6fd2”). Para impedir este mensaje de advertencia, debe importar la clave digital de Fedora en el anillo de claves de su sistema. Teclee

```
[root@serverA ~]# rpm --import /usr/share/rhn/RPM-GPG-KEY-fedora
```

También podría tener que importar las claves de otros vendedores en el anillo de claves. Para tener la certeza adicional de que la clave local que usted tiene no es una falsificación, puede importar directamente la clave del sitio Web del vendedor. Por ejemplo, para importar una clave del sitio del proyecto de Fedora, teclearía

```
[root@serverA ~]# rpm --import \
http://download.fedoraproject.org/pub/fedora/linux/core/3/i386/os/RPM-GPG-KEY-fedora
```

Administradores de paquetes GUI RPM

Para quienes les gusta una buena herramienta GUI para que les ayude a simplificar sus vidas, existen varios administradores de paquetes con sistemas de entrada GUI. Realizando todo el trabajo sucio detrás de estos preciosos sistemas de entrada GUI se encuentra RPM. Las herramientas GUI le permiten hacer un buen número de acciones sin forzarlo a recordar parámetros de líneas

de comandos. En las secciones que siguen se da una lista de algunos de los más populares con cada distribución o entorno de escritorio.

Fedora

Puede lanzar la herramienta GUI de administración de paquetes (figura 3-1) en Fedora al hacer clic en Main menu | System Settings | Add/Remove Applications (Menú principal | Ajustes del sistema | Agregar/Eliminar aplicaciones). También puede lanzar el administrador de paquetes Fedora a partir de la línea de comandos sólo con teclear

```
[root@serverA ~]# system-config-packages
```

SuSE

En SuSE, la mayor parte de la administración del sistema se hace a través de una herramienta llamada YaST. Esta última (vea la figura 3-2) está formada por diferentes módulos. Para agregar y eliminar paquetes de manera gráfica en el sistema, el módulo pertinente se llama **sw_single**. De

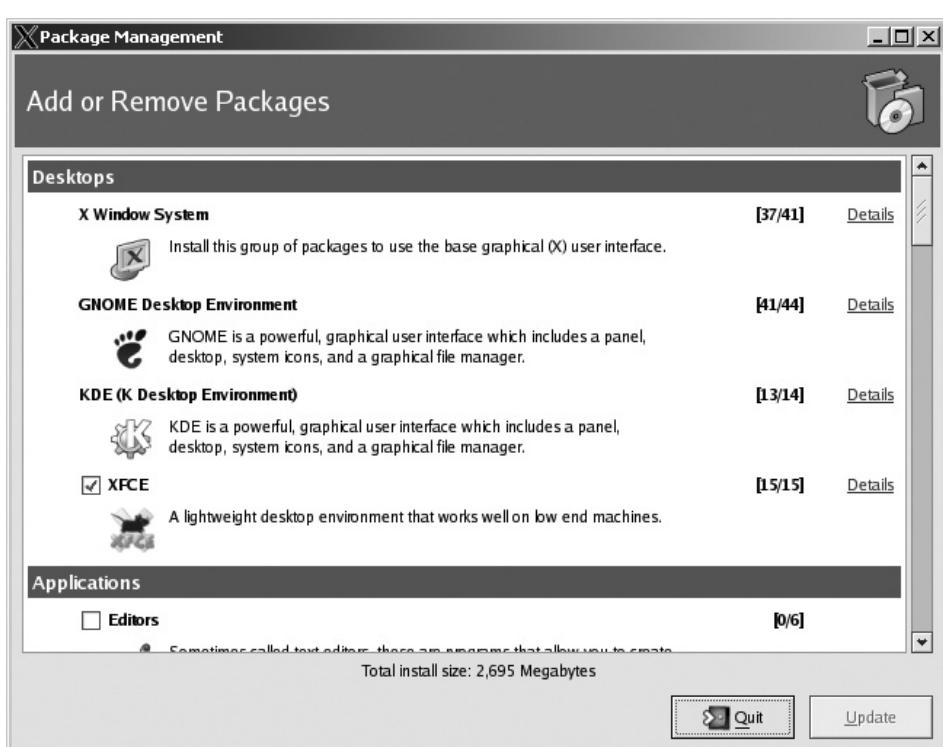


Figura 3-1. Administrador de paquetes GUI de Fedora Core

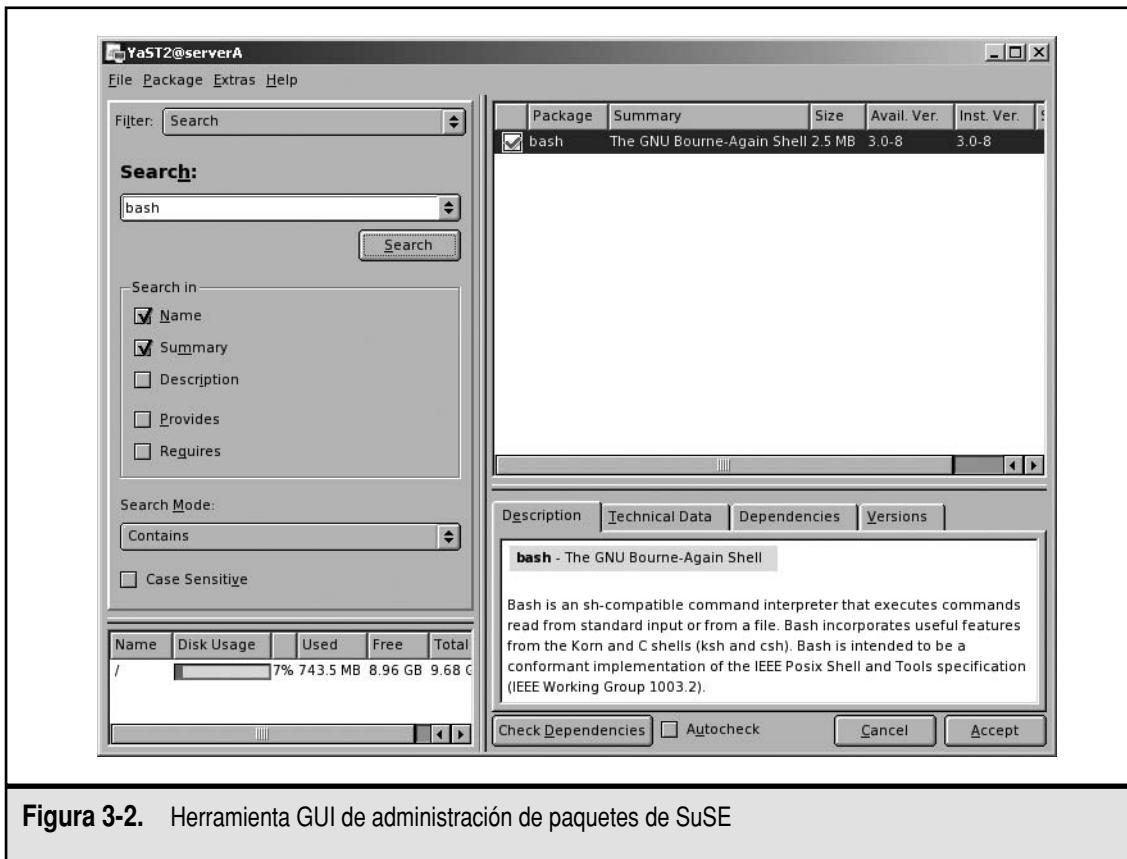


Figura 3-2. Herramienta GUI de administración de paquetes de SuSE

modo que para lanzar este módulo a partir de la línea de comandos de un sistema en el que se esté ejecutando la versión SuSE de Linux, teclearía

```
serverA:~ # yast2    sw_single
```

Yum

Yum es uno de los métodos más recientes de administración del software en los sistemas Linux. Básicamente es un shell para RPM con grandes mejoras. Ha estado por allí durante un tiempo, pero se ha vuelto de uso cada vez más amplio y ha sido adoptado a medida que los principales vendedores de Linux decidieron concentrarse más en sus ofertas de productos comerciales (más redituables). Yum parece ser una de esas tecnologías perturbadoras porque ha cambiado de manera radical el procedimiento tradicional para la administración de paquetes. Los grandes sitios populares que sirven como depositarios para software de fuente abierta han tenido que reestructurar un poco las herramientas para mantener y dar lugar ahora también a los depositarios "Yumificados."

Según la página Web del proyecto Yum, “Yum es un actualizador automático e instalador/eliminador de paquetes para sistemas rpm. Calcula en forma automática las dependencias y deduce lo que debe ocurrir para instalar los paquetes. Facilita el mantenimiento de los grupos de máquinas, sin tener que actualizar en forma manual cada una usando rpm”.

Este resumen se queda corto. Yum puede hacer mucho más que eso. Existen nuevas distribuciones de Linux que se apoyan con intensidad en las capacidades que Yum proporciona.

Usar Yum es muy sencillo en los sistemas soportados. En su mayor parte, usted necesita un solo archivo de configuración (`/etc/yum.conf`) que apunta hacia un depositario de software habilitado con Yum (yumificado). Por fortuna, varias distribuciones de Linux ahora se embarcan con Yum ya instalado y preconfigurado. Fedora Core es una de esas distribuciones.

Para usar Yum en un sistema Fedora Core, por ejemplo, instalar un paquete llamado “gcc”, a partir de la línea de comandos, teclearía

```
[root@serverA ~]# yum install gcc
```

Yum se encargará en forma automática de cualesquier dependencias que podría necesitar el paquete e instala éste para usted (la primera vez que se ejecuta, estructurará su caché local).

Yum también tiene extensas capacidades que le ayudarán a hallar un paquete, incluso si no conoce el nombre correcto de éste. Todo lo que usted necesita saber es una parte del nombre. Por ejemplo, si desea buscar todos los paquetes que tiene la palabra “header” en el nombre, puede intentar una opción Yum como ésta:

```
[root@serverA ~]# yum search headers
```

Ésta le dará como respuesta una larga lista de correspondencias. Entonces puede recorrer la lista y tomar el paquete que desea.

NOTA De manera predeterminada Yum intenta tener acceso a los depositarios que están ubicados en alguna parte en Internet. Por lo tanto, su sistema necesita poder tener acceso a esta última para usarlo en su estado predeterminado. También, usted siempre puede crear su propio depositario local de software, en el sistema local de archivos o en su LAN, y yumificarlo. Sencillamente, copie el contenido completo de los medios de distribución (DVD/CD) en alguna parte y ejecute el comando `yum -arch` contra la ubicación del directorio.

COMPILE E INSTALE EL SOFTWARE GNU

Uno de los beneficios clave del software de fuente abierta es que usted tiene acceso al código fuente. Si el desarrollador decide dejar de trabajar en él, usted puede continuar (si sabe cómo hacerlo). Si encuentra un problema, puede resolverlo. En otras palabras, tiene la situación bajo control y no está a merced de un desarrollador comercial que usted no puede controlar. Pero tener el código fuente significa que también necesita ser capaz de compilarlo. De lo contrario, todo lo que tiene es un montón de archivos de texto que no pueden hacer mucho.

Aun cuando casi todas las piezas de software mencionadas en este libro se encuentran como un RPM, recorremos los pasos de compilación y estructuración del software a partir de la fuente. Esto tiene el beneficio adicional de permitirle tomar y elegir las opciones de tiempo de compilación, lo cual es algo que no puede hacer con los RPM preestructurados. Asimismo, podría compilarse un RPM para una arquitectura específica, como la Intel 486, pero ese mismo código podría ejecutarse mejor si la compila en su forma nativa en su, digamos, CPU clase Intel Pentium 4.

En esta sección, recorremos los pasos del proceso de compilación del paquete Hello, un paquete de software GNU que al principio podría parecer inútil pero que existe por buenas razones. La mayor parte del software GNU se conforma a un método estándar de compilación e instalación, y el software "hello" intenta conformarse a este estándar y, por consiguiente, constituye un ejemplo excelente.

Obtención y desempaque del paquete

La otra relación les dejó un mal sabor de boca, pero están listos para intentarlo una vez más. Quizá las cosas no funcionaron lo bastante bien en virtud de que había que tratar con muchos otros factores...RPM con sus opciones sin fin y su sintaxis aparentemente intrincada. Y, por consiguiente, fuera lo viejo y vamos con lo nuevo. Puede ser que usted tenga más suerte esta vez, si tiene más control sobre el flujo de las cosas.

Aunque un poco más complicado, el trabajo directo con el código fuente le dará más control sobre el software y sobre la manera en que se conforman las cosas.

El software que viene en la forma fuente por lo general se pone a disposición como una *tarball* (bola de brea); es decir, se archiva en un solo archivo grande y, a continuación, se comprime. Las herramientas de uso común para realizar esto son **tar** y **gzip**: **tar** maneja el proceso de combinar muchos archivos en un solo archivo grande y **gzip** es la responsable de la compresión.

NOTA No confunda el programa **gzip** de Linux con el programa WinZip de MS Windows. Son dos programas diferentes en los que se usan dos métodos distintos (pero comparables) de compresión. El programa **gzip** de Linux puede manejar archivos que se comprimen por medio de WinZip y este último programa sabe cómo tratar con *tarballs*.

Tratemos de instalar hello, un paso a la vez:

1. En primer lugar, empecemos por obtener una copia del código fuente hello.

NOTA Por lo común se selecciona un solo directorio en el cual estructurar y almacenar las *tarballs*. Esto permite que el administrador del sistema conserve la *tarball* de cada paquete en un lugar seguro en el caso de que necesite posteriormente retirar algo de ella. También permite a todos los administradores saber cuáles paquetes se instalan en el sistema además del sistema base. Un buen directorio para esto es **/usr/local/src**, ya que, por lo general, el software local para un sitio se instala en **/usr/local**.

Baje una copia del programa hello que se usa en este ejemplo de <http://www.gnu.org/software/hello> o, directamente, de <http://ftp.gnu.org/gnu/hello/hello-2.1.1.tar.gz>.

La versión más reciente del programa de la que se disponía en el momento en que se escribió esto era **hello-2.1.1.tar.gz**. Guarde el archivo en el directorio **/usr/local/src/**.

SUGERENCIA Una manera rápida de descargar un archivo de Internet (a través de FTP o HTTP) es usando el programa utilitario llamado **wget**. Por ejemplo, para bajar el programa hello mientras se encuentra en un mensaje del shell, sencillamente teclearía:

```
# wget http://ftp.gnu.org/gnu/hello/hello-2.1.1.tar.gz
```

Y el archivo se guardará en forma automática en su directorio actual de trabajo (PWD, present working directory).

- Después de descargar el archivo, necesitará desempacarlo (o **untar**). Al desempacar, una *tarball* en general creará un nuevo directorio para todos sus archivos. Por ejemplo, la *tarball* hello (**hello-2.1.1.tar.gz**) crea el subdirectorio **hello-2.1.1**. La mayor parte de los paquetes siguen este estándar. Si encuentra un paquete que no lo siga, es una buena idea crear un subdirectorio con un nombre razonable y colocar allí todos los archivos fuente desempacados. Esto permite que ocurran múltiples estructuraciones al mismo tiempo, sin el riesgo de que dos estructuras entren en conflicto. Use el comando **tar** para desempacar y descomprimir el archivo hello. Teclee

```
[root@serverA src]# tar -xvzf hello-2.1.1.tar.gz
hello-2.1.1/
hello-2.1.1/intl/
hello-2.1.1/intl/ChangeLog
hello-2.1.1/intl/Makefile.in
hello-2.1.1/intl/locale.alias
hello-2.1.1/intl/ref-add.sin
```

El parámetro **z** en este comando **tar** hace que **gzip** descomprima el archivo antes de que ocurra el proceso **untar**. El parámetro **v** le dice a **tar** que muestre el nombre del archivo que se está sujetando a **untar**, a medida que se desarrolla el proceso. De esta manera, usted conocerá el nombre del directorio en donde se están desempacando todas las fuentes.

NOTA Podría encontrar archivos que finalizan con la extensión **.tar.bz2**. Bzip2 es un algoritmo de compresión que está ganando popularidad y **tar** de GNU sí soporta descomprimirllo en la línea de comandos con la opción **y** o **j** (en lugar del parámetro **z**).

- Debe haberse creado para usted un nuevo directorio llamado **hello-2.1.1** en el transcurso del proceso **untar**. Cambie el nuevo directorio y haga una lista de su contenido. Teclee

```
[root@serverA src]# cd hello-2.1.1 ; ls
```

Búsqueda de la documentación (empezar a conocerse entre sí; una vez más)

Bueno. Ahora ya están los dos descargados...perdón, se han encontrado. Es probable que ahora sea un buen momento para mirar alrededor y ver si viene con alguna documentación especial . . . perdón, con necesidades especiales.

Un buen lugar para buscar documentación para el software será en la raíz de su árbol de directorios. Una vez que esté dentro del directorio con todo el código fuente, empiece a buscar la documentación. *¡Lea siempre la documentación que viene con el código fuente!* Si existen algunas direcciones especiales para la compilación, notas o advertencias, lo más probable es que se mencionen aquí. Se ahorrará un gran esfuerzo si lee en primer lugar los archivos pertinentes.

Entonces, por consiguiente, ¿cuáles son los archivos pertinentes? Lo normal es que estos archivos tengan nombres como **README (LÉAME)** e **INSTALL (INSTALACIÓN)**. Puede ser que el desarrollador también haya puesto toda la documentación disponible en un directorio nombrado de manera apropiada como **docs**.

En general, el archivo **README** incluye una descripción del paquete, referencias a documentación adicional (incluyendo la documentación de instalación) y referencias relativas al autor del paquete. Lo normal es que el archivo **INSTALL** tenga direcciones para compilar e instalar el paquete.

Por supuesto, éstos no son absolutos. Cada paquete tiene sus peculiaridades. La mejor manera de averiguarlo es sencillamente hacer una lista del contenido del directorio y buscar los signos obvios de documentación adicional. En algunos paquetes se usan de manera diferente las mayúsculas: **readme**, **README**, **ReadMe**, y así sucesivamente. (Recuerde, Linux diferencia mayúsculas y minúsculas!) Algunos introducen variaciones sobre un tema, como **README.1ST**, o bien, **README.NOW**, etcétera.

Mientras se encuentra en el directorio **/usr/local/src/hello-2.1.1**, use un buscador para ver el archivo **INSTALL** que viene con el programa hello. Teclee

```
[root@serverA hello-2.1.1]# less INSTALL
```

Salga del buscador tecleando **q** cuando haya terminado de leer el archivo.

SUGERENCIA Otro buscador popular que puede usar en lugar de **less** se llama **more!** (Nota histórica: **more** vino antes que **less**.)

Configuración del paquete

Ambos quieren esta relación para trabajar y es posible durante más tiempo que en los casos anteriores. De modo que éste es el momento de establecer las directrices y las expectativas.

La mayor parte de los paquetes se embarcan con un script de autoconfiguración; resulta seguro suponer que así es, a menos que su documentación diga lo contrario. Por lo general, a estos scripts se les nombra **configure** (o **config**) y pueden aceptar parámetros. Se tiene un corto número de parámetros en existencia, de los que se dispone a través de todos los scripts **configure**, pero lo interesante ocurre en términos de programa a programa. Cada paquete tendrá un manojo de características que se pueden habilitar o deshabilitar o que tienen valores especiales en el momento de la compilación, y deben estructurarse a través de **configure**.

Para ver cuáles opciones de **configure** vienen con un paquete, sencillamente ejecute

```
[root@serverA hello-2.1.1]# ./configure --help
```

Sí, se tienen dos guiones (--) antes de la palabra "help".

NOTA Una opción de la que se dispone en forma común es `--prefix`. Esta opción le permite fijar el directorio base en donde el paquete quede instalado. De manera predeterminada, en la parte de los paquetes se usa `/usr/local`. Cada componente del paquete se instalará en el directorio apropiado en `/usr/local`.

Si está contento con las opciones predeterminadas que le ofrece el script `configure`, teclee

```
[root@serverA hello-2.1.1]# ./configure  
verificando en busca de una instalación compatible BSD... /usr/bin/install -c  
comprobando si el entorno de estructuración no está dañado... sí  
comprobando la existencia de gawk... gawk  
comprobando si se hacen conjuntos ${MAKE}... sí  
comprobando la existencia de gcc... gcc  
comprobando la existencia de la salida predeterminada del compilador C...  
a.out  
...<SALIDA TRUNCADA>...  
estado de la configuración: ejecutando los comandos predeterminados
```

Con todas las opciones que deseé estructurar, una ejecución del script `configure` creará un tipo especial de archivo llamado *makefile* (archivo de producción). Los makefiles constituyen la base de la fase de compilación. En general, si `configure` falla, usted no obtendrá un makefile. Asegúrese de que el comando `configure` en efecto se completó sin errores.

Compilación del paquete

Esta etapa en realidad no se ajusta bien en ninguna parte de nuestro modelo de salir con alguien. Pero podría considerarla como semejante a ese periodo en el que está completamente ciego de amor y por ello todo sólo flota, y muchas cosas sólo son inexplicables.

Todo lo que necesita hacer es ejecutar `make`, como sigue:

```
[root@serverA hello-2.1.1]# make
```

La herramienta `make` lee todos los *makefiles* que el script `configure` creó. Estos archivos le dicen a `make` cuáles archivos compilar y el orden en el cual tiene que compilarlos, lo cual es crucial, ya que podría haber cientos de archivos fuente.

Dependiendo de la velocidad de su sistema, la memoria disponible y cuán ocupado esté su sistema haciendo otras cosas, el proceso de compilación podría tardar algo de tiempo en completarse, de modo que no se sorprenda.

Conforme `make` esté trabajando, presentará cada comando que esté ejecutando y todos los parámetros asociados con él. Esta salida suele ser la invocación del compilador y de todos los parámetros pasados a éste; es algo bastante tedioso, ¡en tal forma que incluso los programadores se inclinaron a automatizarlo!

Si la compilación se desarrolla sin incidentes, no verá mensajes de error. La mayor parte de los mensajes de error del compilador son muy claros y distintos, de modo que no se preocupe respecto a la posibilidad de pasar por alto un error. Si en realidad ve un error, no entre en pánico. La mayor parte de los mensajes de error no reflejan un problema con el propio programa, sino por lo común con el sistema de alguna manera u otra. Lo normal es que estos mensajes sean el resultado de permisos inapropiados de los archivos, o bien, de archivos que no pueden encontrar.

En general, tranquilícese y lea el mensaje de error. Incluso si el formato es un poco raro, puede explicar lo que está mal en lenguaje sencillo, permitiendo en consecuencia que usted lo arregle con rapidez. Si el error todavía es confuso, mire la documentación que vino con el paquete para ver si tiene una lista de correo o una dirección de correo electrónico con la que pueda ponerse en contacto para pedir ayuda. La mayor parte de los desarrolladores se ponen más que contentos por proporcionar ayuda, pero usted necesita recordar ser amable e ir al punto (en otras palabras, no inicie su correo despotricando acerca de por qué su software es terrible).

Instalación del paquete

Ha hecho casi todo lo demás. Ha encontrado sus socios, los ha estudiado, incluso los ha compilado, ahora es el momento de moverlos con usted.

A diferencia de la etapa de compilación, lo normal es que la de instalación se desarrolle con suavidad. En la mayor parte de los casos, una vez que se completa con éxito la compilación, todo lo que necesita hacer es ejecutar

```
[root@serverA hello-2.1.1]# make install
```

Con esto se instalará el paquete en la ubicación especificada por el argumento de prefijo predeterminado (**--prefix**) que se usó con anterioridad con el script **configure**.

Se iniciará el script de instalación (que suele estar empotrado en el makefile). Debido a que **make** presenta cada comando a medida que se está ejecutando, verá por ello una gran cantidad de texto volando. No se preocupe por ello; es perfectamente normal. A menos que vea un mensaje de error, el paquete se instala.

Si en realidad ve un mensaje de error, lo más probable es que se deba a problemas de permisos. Vea el último archivo que se estuvo tratando de instalar antes de la falla y, a continuación, vaya a revisar todos los permisos requeridos para colocar un archivo allí. Puede ser que necesite usar los comandos **chmod**, **chown** y **chgrp** para este paso.

SUGERENCIA Si se tiene la intención de que el software que se está instalando es para que se use y esté disponible para todo el sistema, ésta es casi siempre la etapa que necesita ser realizada por el superusuario (es decir, raíz). En consecuencia, la mayor parte de las instrucciones de instalación requerirán que usted se convierta en la raíz, antes de realizar este paso. Si, por otra parte, un usuario común está compilando e instalando un paquete de software para su propio uso personal, en un directorio para el cual ese usuario tiene plenos permisos (por ejemplo, especificando **--prefix=/home/user_name**), entonces no hay necesidad de convertirse en raíz para hacer esto.

Prueba del software

Un error común que los administradores cometan es pasar por todo el proceso de configuración y compilación y, a continuación, cuando instalan, no probar el software para asegurarse que se ejecuta como debe ser. Siendo un usuario común, también es necesario que se realice la prueba del software, si éste va a ser usado por usuarios que no son raíz. En nuestro ejemplo, ejecute el comando **hello** para verificar que los permisos están correctos y que los usuarios no tendrán problemas en la ejecución del programa. Usted cambiará con rapidez los usuarios (usando el comando **su**) para tener la seguridad de que cualquiera puede usar el software.

Suponiendo que aceptó el prefijo predeterminado de instalación para el programa hello (es decir, los archivos pertinentes estarán bajo el directorio `/usr/local`), use la trayectoria completa hacia el binario del programa para ejecutarlo. Teclee

```
[root@serverA hello-2.1.1]$ /usr/local/bin/hello  
¡Hola, mundo!
```

Es todo; ha terminado.

Limpieza

Una vez que se instala el paquete, puede realizar algo de limpieza para deshacerse de todos los archivos temporales creados en el transcurso de la instalación. Puesto que tiene la tarball original del código fuente, está bien que se deshaga de todo el directorio a partir del cual compiló ese código. En el caso del programa hello, se desharía de `/usr/local/src/hello-2.1.1`.

Empiece por ir a un nivel de directorio arriba del que desea eliminar. En este caso, sería `/usr/local/src`.

```
[root@serverA hello-2.1.1]# cd /usr/local/src
```

Ahora use el comando `rm` para eliminar el directorio actual, como sigue:

```
[root@serverA hello-2.1.1] rm -rf hello-2.1.1
```

El comando `rm`, en especial con el parámetro `-rf`, es muy peligroso. Elimina de manera recurrente un directorio completo, sin detenerse a verificar alguno de los archivos. Es especialmente potente cuando lo ejecuta el usuario raíz; primero le disparará y después dejará que haga las preguntas.

Sea cuidadoso y asegúrese que se está borrando lo que usted quiere borrar. No existe manera fácil de deshacer la eliminación de un archivo en Linux, cuando se trabaja a partir de la línea de comandos.

PROBLEMAS COMUNES EN LA ESTRUCTURACIÓN A PARTIR DE LA FUENTE

El programa hello de GNU podría no tener la apariencia de una herramienta muy útil y, en su mayor parte, estamos de acuerdo en que no lo es. Pero algo valioso que proporciona es la capacidad de probar el compilador en su sistema. Si usted acaba de finalizar la tarea de actualizar su compilador, compilar este sencillo programa le suministrará una comprobación de sanidad de que, en efecto, el compilador está funcionando.

Enseguida se dan algunos otros problemas (y sus soluciones) con los que puede encontrarse al estructurar a partir de la fuente.

Problemas con las bibliotecas

Un problema contra el que podría chocar es cuando el programa no puede hallar un archivo del tipo “`libsomething.so`” y, por esa razón, termina. Este archivo es lo que se conoce como *biblioteca*

(library). Las bibliotecas son los sinónimos de los DDL en Windows. Estas bibliotecas se almacenan en varios lugares en el sistema Linux y, por lo común, residen en `/usr/lib` y `/usr/local/lib`. Si ha instalado un paquete de software en un lugar diferente de `/usr/local`, tendrá que configurar su sistema o su shell para saber en dónde buscar estas nuevas bibliotecas.

NOTA Las bibliotecas de Linux se pueden ubicar en cualquier parte en su sistema de archivos. Apreciará la utilidad de esto cuando, por ejemplo, tenga que usar el Network File System (NFS, Sistema de archivos de la red) para compartir un directorio (o, en nuestro caso, software) entre los clientes de la red. Encontrará que los usuarios o clientes pueden usar con facilidad el software que reside en el compartimiento de la red.

Existen dos métodos para configurar las bibliotecas en un sistema Linux. Uno de ellos es modificar `/etc/ld.conf`, agregar la trayectoria de sus nuevas bibliotecas y usar el comando `ldconfig -m`, para cargar en los nuevos directorios. También puede usar la variable de entorno `LD_LIBRARY_PATH` a fin de mantener una lista de directorios de bibliotecas para buscar archivos de estas últimas. Lea la página man en busca de `ld.conf`, para obtener más información.

Cuando no hay script para configurar

A veces, descargará un paquete y, al instante, `cd` en un directorio, y ejecutará `./configure`. Y probablemente recibirá un impacto cuando lea el mensaje “No such file or directory” (No existe ese archivo o directorio). Como se dijo con anterioridad en el capítulo, lea los archivos `README` e `INSTALL` en la distribución. Lo normal es que los autores del software sean suficientemente corteses para proporcionarle por lo menos estos dos archivos. Es muy fácil querer saltar derecho hacia dentro y empezar a compilar algo sin mirar primero los documentos y, después, regresar horas más tarde para hallar que se pasó por alto un paso. El primer paso a dar al instalar software es leer la documentación. Probablemente le señalará el hecho de que necesita ejecutar primero `imake` y, enseguida, `make`. Tenga la idea: siempre lea primero la documentación y, después, proceda a compilar el software.

Código fuente interrumpido

No importa lo que haga, es posible que el código fuente esté sencillamente interrumpido y la única persona que puede hacerlo trabajar o hacer que tenga algún sentido es su autor original. Muchas veces, puede ser que haya consumido incontables horas intentando hacer que la aplicación compile y estructure, antes de llegar a esta conclusión y lanzar la toalla. También es posible que el autor del programa haya dejado sin documentar información valiosa y pertinente.

RESUMEN

Ha explorado el uso del popular administrador de paquetes RPM. Utilizó sus diversas opciones para manipular los paquetes RPM, consultando, instalando y desinstalando un paquete muestra. También exploró con brevedad las otras opciones para usar `rpm` de modo directo a partir de la línea de comandos. Las opciones son en su mayor parte las diversas herramientas GUI de las que se dispone. Las herramientas GUI son muy semejantes a la miniaplicación Windows Add/Remo-

ve Programs Control Panel (Panel de control, Agregar/Eliminar programas de Windows). Sólo apunte y haga clic. También tocamos con brevedad un sistema de administración de software, ahora muy popular, en Linux llamado Yum.

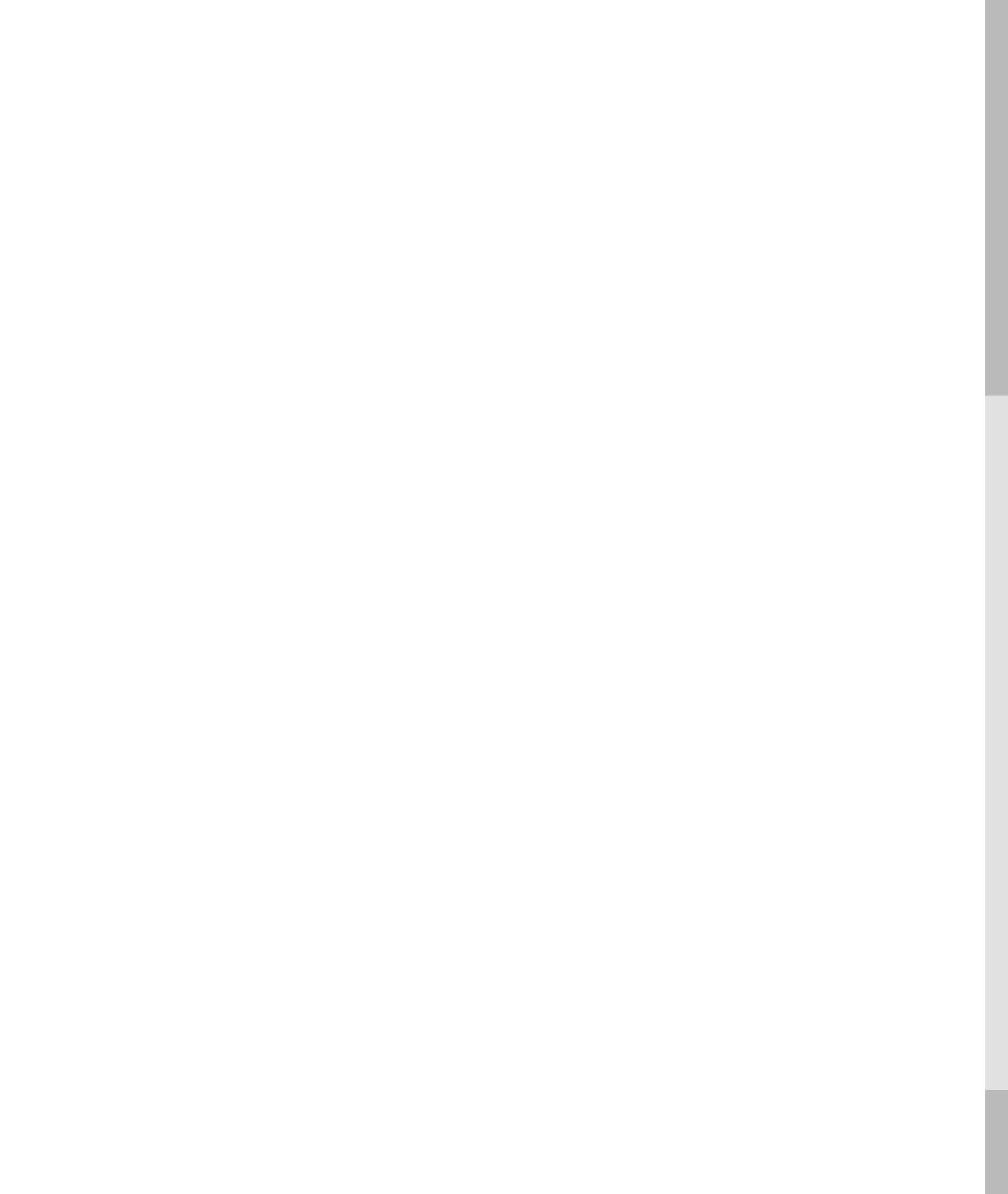
Usando un programa disponible de fuente abierta como ejemplo, se describieron los pasos que intervienen en la configuración, compilación y estructuración de software a partir del código fuente sin procesar.

Como beneficio adicional, también aprendió una cosa o dos acerca de la mecánica de las relaciones.

PARTE II



Administración
con un solo anfitrión



CAPÍTULO 4



Administración
de los usuarios

UNIX/Linux se diseñó desde las bases hasta convertirse en un sistema operativo para múltiples usuarios. Un sistema operativo de este tipo no será de mucha utilidad sin usuarios. Y esto nos lleva al tema de administración de los usuarios en Linux. Asociado con cada usuario se encuentra equipaje de cada uno. Este equipaje podría incluir archivos, procesos, recursos y otra información. Al tratar con un sistema de múltiples usuarios, para un administrador de sistemas es necesario que haya comprendido bien lo que constituye un usuario (y todo lo que es el equipaje de éste), un grupo y cómo interactúan estos entre sí.

En los sistemas de computadoras, se usan las cuentas de usuarios con el fin de determinar quién tiene acceso a qué. La capacidad de un usuario para tener acceso a un sistema se determina por medio de si existe o no ese usuario y si tiene los permisos apropiados para usar tal sistema.

En este capítulo, examinaremos la técnica para administrar los usuarios en un solo anfitrión. Empezaremos por examinar los archivos actuales de base de datos que contienen la información acerca de los usuarios. De allí, examinaremos las herramientas del sistema de las que se dispone para administrar los archivos en forma automática.

¿QUÉ CONSTITUYE EXACTAMENTE UN USUARIO?

En Linux, cada archivo y cada programa debe ser propiedad de un *usuario*. Cada usuario tiene un identificador único llamado la *ID del usuario (UID)*. También, cada usuario debe pertenecer por lo menos a un *grupo*, una colección de usuarios establecida por el administrador del sistema. Los usuarios pueden pertenecer a varios grupos. Como los usuarios, los grupos también tienen identificadores únicos, llamados *ID de los grupos (GID)*.

La accesibilidad de un archivo o programa se basa en sus UID y GID. Un programa en ejecución hereda los derechos y permisos del usuario que lo llama (con SetUID y SetGID, los cuales se discuten en “Comprendión de los programas SetUID y SetGID” más adelante en este capítulo, se crea una excepción a esta regla). Los derechos de cada usuario se pueden definir en una de dos maneras: como los de un *usuario normal* y los de un *usuario raíz*. Los usuarios normales sólo pueden tener acceso a lo que poseen o a lo que se les ha dado permiso de ejecutar; el permiso se concede porque el usuario pertenece al grupo del archivo o porque el archivo es accesible a todos los usuarios. A los usuarios raíz se les permite el acceso a todos los archivos y programas del sistema, sin importar si la raíz les pertenece o no. A menudo, al usuario raíz se le conoce como *superusuario*.

Si usted está acostumbrado a Windows, puede hallar un paralelismo entre la administración de usuarios del sistema y la administración de usuarios de Linux. Por ejemplo, las UID son comparables a las SID (ID del sistema) de Windows. Contrastando con Windows NT, puede hallar el modelo de seguridad de Linux simplista de manera exasperante: usted es usuario raíz o no lo es. Los usuarios normales no pueden tener los privilegios de los raíz, de la misma manera que, en NT, a ese tipo de usuarios no se les puede conceder el acceso del administrador. Aun cuando este enfoque es un poco menos común, en Linux también puede implementar el control del acceso de grano más fino a través del uso de las listas de control de acceso (ACL), como lo puede hacer con Windows. ¿Cuál sistema es mejor? Depende de lo que usted desea y de a quién le pregunte.

Dónde se guarda la información del usuario

Si el lector ya ha usado hasta la administración de los usuarios de Windows 2000, está familiarizado con la herramienta Active Directory (Directorio activo) que se encarga de los detalles sustanciales de la base de datos de los usuarios. Esta herramienta resulta conveniente, pero dificulta el desarrollo de sus propias herramientas administrativas, dado que la única otra manera de leer o manipular la información de los usuarios es a través de una serie de LDAP, Kerberos o llamadas programáticas del sistema.

Como contraste, Linux toma el camino del UNIX tradicional y conserva toda la información de los usuarios en archivos directos de texto. Esto resulta benéfico por la sencilla razón de que permite hacer cambios a la información de los usuarios, sin necesidad de alguna otra herramienta que no sea un editor de textos, como **vi**. En muchos casos, los sitios más grandes sacan ventaja de estos archivos de texto al desarrollar sus propias herramientas de administración de los usuarios, de modo que no sólo pueden crear cuentas nuevas sino también hacer adiciones en forma automática al directorio telefónico corporativo, a las páginas Web, etcétera.

Sin embargo, es posible que los usuarios y grupos que trabajan con el estilo de UNIX por primera vez prefieran adherirse a las herramientas básicas de administración de los usuarios que vienen con la distribución de Linux. Más adelante, en este capítulo, discutiremos esas herramientas en "Herramientas para administración de los usuarios". Por ahora, examinemos los archivos de texto en los que se almacena la información de los usuarios y los grupos, en Linux.

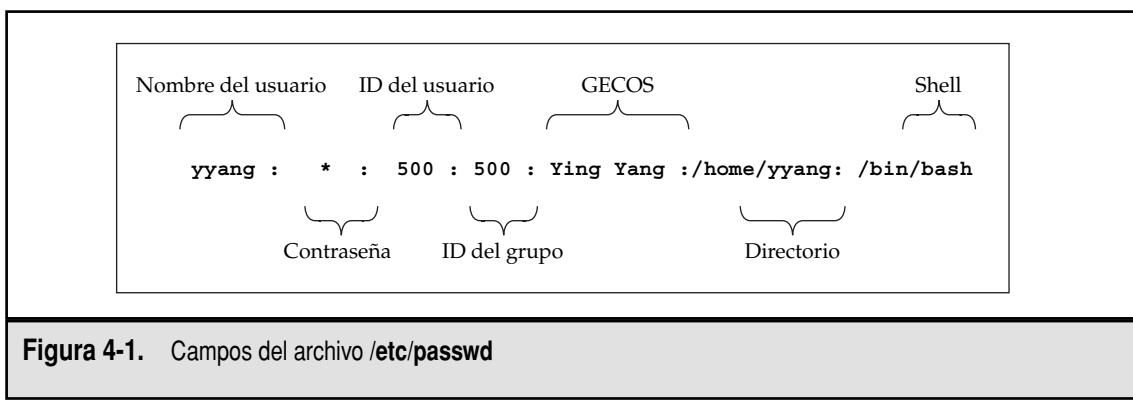
El archivo /etc/passwd

En el archivo **/etc/passwd** se almacena la concesión de acceso (login) al usuario, la entrada cifrada de la contraseña, la UID, la GID predeterminada, el nombre (a veces llamado GECOS), el directorio inicial y el shell de la concesión de acceso. Cada línea en el archivo representa información acerca del usuario. Las líneas se forman con varios campos estándar, delimitándose cada campo por medio de dos puntos. En la figura 4-1, se ilustra una entrada muestra de un archivo **passwd**, con sus diversos campos.

En las secciones que siguen, se discuten con detalle los campos del archivo **/etc/passwd**.

Campo del nombre del usuario

Este campo también se conoce como el de concesión de acceso o de la cuenta. En él se almacena el nombre del usuario en el sistema. El nombre del usuario debe ser una cadena única y que identifica también de manera única un usuario del sistema. Los diferentes sitios usan métodos distintos para generar los nombres de concesión de acceso de los usuarios. Un método muy común es usar la primera letra del nombre (o primer nombre) del usuario y anexar el apellido (o apellidos) de éste. Por lo común, esto funciona porque las posibilidades son relativamente remotas de que se tengan usuarios con el mismo nombre y el mismo apellido (o mismos apellidos). Desde luego, existen diversas variaciones de este método. Por ejemplo, para un usuario cuyo nombre es "Ying" y su apellido es "Yang", se le puede asignar un nombre de usuario de "yyang".



Campo de la contraseña

Este campo contiene la contraseña cifrada del usuario. En la mayor parte de los sistemas Linux modernos, este campo contiene una letra *x* para indicar que, en el sistema, se están usando contraseñas sombra (lo que se discute con detalle más adelante). Todas las cuentas de usuarios en el sistema deben tener una contraseña o, al menos, etiquetarlos como imposibles para dejarlos entrar. Esto es crucial para la seguridad del sistema; las contraseñas débiles hacen que un sistema sea comprometedor, sólo que mucho más sencillo.

En realidad, la filosofía que se encuentra detrás de las contraseñas es bastante interesante, en especial porque en la actualidad todavía dependemos en una parte significativa de ella. La idea es sencilla: en lugar de confiar en archivos protegidos para mantener las contraseñas en secreto, el sistema cifraría la contraseña utilizando un algoritmo desarrollado por AT&T (y aprobado por la National Security Agency, Agencia Nacional de Seguridad) y que se conoce con el nombre de Data Encryption Standard (DES, Norma de cifrado de datos) y deja el valor cifrado a la vista del público. Lo que originalmente hizo que esto fuera seguro era que el algoritmo de cifrado era difícil de descifrar por medio de la computación. Lo más que pudo hacer la mayor parte de los muchachos fue un ataque a fuerza bruta al diccionario, en donde sistemas automatizados realizarían repeticiones de uno a otro lado de un gran diccionario y se atendrían a la naturaleza de los usuarios de tomar palabras del idioma como sus contraseñas. Mucha gente trató de descifrar el propio DES, pero como era un algoritmo abierto que cualquiera podía estudiar, se hizo mucho más a prueba de balas, antes de que en realidad se desplegara.

Cuando los usuarios hicieran entrar sus contraseñas en un mensaje para solicitar el acceso, esa contraseña se cifraría. A continuación, el valor cifrado se compararía contra la entrada de contraseña del usuario. Si los dos valores cifrados coincidían, se permitiría que el usuario entrara al sistema. El algoritmo real para realizar el cifrado era, desde el punto de vista computacional, suficientemente fácil que un solo cifrado no sería demasiado tardado. Sin embargo, las decenas de miles de cifrados que se necesitarían para un ataque al diccionario sería prohibitivamente tardado.

Pero entonces se presentó un problema: se cumplió la ley de Moore sobre la duplicación de la velocidad del procesador cada 18 meses y las computadoras domésticas se volvieron lo bastante poderosas y rápidas que los programas fueron capaces de realizar un ataque a fuerza bruta al diccionario en el transcurso de días, en lugar de semanas o meses. Los diccionarios se hicieron más grandes y el software, más inteligente. Por consiguiente, fue necesario volver a evaluar la naturaleza de las contraseñas. Una solución ha sido mejorar el algoritmo empleado para realizar el cifrado de las contraseñas. En algunas distribuciones de Linux se sigue la trayectoria trazada por el sistema operativo FreeBSD y se utiliza el esquema MD5. Esto ha aumentado la complejidad relacionada con el descifrado de las contraseñas, lo cual, cuando se usa en conjunción con las contraseñas sombra (que se discuten más adelante), funciona bastante bien. (Por supuesto, ¡esto es suponiendo que usted logra que sus usuarios elijan contraseñas buenas!)

SUGERENCIA La elección de contraseñas buenas siempre es una tarea. De manera inevitable, sus usuarios preguntarán: "Entonces, ¡Oh Todopoderoso Administrador del Sistema!, ¿con qué se forma una buena contraseña?". Aquí tiene su respuesta: una palabra que no sea de un idioma (no inglés, no español, no alemán, no una palabra del lenguaje humano), de preferencia con mayúsculas, minúsculas, números y puntuación mezclados; en otras palabras, una cadena que tenga el aspecto de un ruido de la línea. Bien, esto es del todo bonito y maravilloso, pero qué pasa si una palabra es demasiado difícil para recordar, la mayor parte de la gente hará pedazos su propósito al escribirla y conservarla en un lugar que se vea con facilidad. De modo que lo mejor es ¡hacerla fácil de memorizar! Una buena técnica podría ser elegir una frase y, a continuación, tomar la primera letra de cada palabra de ella. Por tanto, la frase "coffee is VERY GOOD for you and me" se convierte en c!VG4yam. La frase se puede memorizar, incluso si la contraseña resultante no lo es.

Campo de la ID del usuario (UID)

En este campo se almacena un número único que el sistema operativo y otras aplicaciones usan para identificar al usuario y determinar los privilegios de acceso. Es el equivalente numérico del campo del nombre del usuario. La UID debe ser única para cada usuario, con la excepción de la UID 0 (cero). Cualquier usuario que tiene una UID de 0 tiene acceso raíz (administrativo) y, como consecuencia, la plena ejecución del sistema. Por lo común, el único usuario que tiene esta UID específica tiene la raíz de concesión de acceso. Se considera una mala práctica permitir que cualesquiera otros usuarios o nombres de usuarios tengan una UID de 0. Esto difiere de manera notable de los modelos de Windows NT y 2000, en los cuales cualquier número de usuarios puede tener privilegios administrativos.

A veces, distribuciones diferentes de Linux adoptan esquemas distintos de numeración UID. Por ejemplo, en Fedora y RHEL, se reserva la UID 99 para el usuario “nadie”, en tanto que en Linux de SuSE se usa la UID 65534 para el usuario “nadie”.

Campo de la ID del grupo (GID)

El campo siguiente en el archivo `/etc/passwd` es la entrada de la ID del grupo. Es el equivalente numérico del grupo primario al que pertenece el usuario. Este campo también desempeña un papel importante en la determinación de los privilegios de acceso del usuario. Se debe hacer notar que, además de un grupo primario del usuario, un usuario puede pertenecer también a otros grupos (se encuentra más acerca de esto en la sección “El archivo `/etc/group`”).

GECOS

En este campo se pueden almacenar varios trozos de información de un usuario. Puede actuar como un lugar de reserva para la descripción del usuario, nombre completo (nombre y apellidos), número telefónico, etcétera. Este campo es opcional y, como resultado, se puede dejar en blanco. También es posible almacenar entradas múltiples en este campo, sencillamente al separar las entradas diferentes con una coma.

NOTA GECOS es un acrónimo de General Electric Comprehensive Operating System (Sistema operativo detallado de General Electric) (ahora conocido como GCOS) y es un remanente de los primeros días de la computación.

Como una nota histórica al pie, la liga de GECOS a UNIX se origina en el hecho de que en los Bell Labs, durante la creación de UNIX, se usaron máquinas GCOS para imprimir. Para dar cabida al uso de los servicios de impresión basados en GCOS, se agregó un campo más al archivo `/etc/passwd`.

Directorio

Éste suele ser el directorio inicial del usuario, pero también puede ser cualquier lugar arbitrario en el sistema. Cada usuario que en realidad ingresa al sistema necesita un lugar para los archivos de configuración que son únicos para él. Este lugar, conocido como *directorio inicial*, permite a cada usuario trabajar en un entorno personalizado, sin tener que cambiar el entorno personalizado por otro usuario; incluso si los dos son admitidos en el sistema al mismo tiempo. En este directorio, a los usuarios se les permite conservar no sólo sus archivos de configuración sino también sus archivos de trabajo regular.

Scripts de arranque

En Linux, los scripts de arranque no son del todo parte de la información almacenada en la base de datos de los usuarios. Sin embargo, desempeñan un papel muy importante en la determinación y control del entorno del usuario. En particular, en Linux, los scripts de arranque suelen almacenarse bajo el directorio inicial del usuario...y, por ello, la necesidad de mencionarlos mientras se está tocando el tema del campo directorio (directorio inicial) en el archivo `/etc/passwd`.

Linux/UNIX se estructuró desde su puesta en marcha como un entorno de usuarios múltiples. A cada usuario se le permite tener sus propios archivos de configuración; por tanto, el sistema aparece personalizado para cada usuario en particular (incluso si se admiten otras personas al mismo tiempo). La personalización del entorno de cada usuario por separado se hace a través del uso de los scripts shell, la ejecución de archivos de control y cosas semejantes. Estos archivos pueden contener una serie de comandos que van a ser ejecutados por el shell que arranca cuando a un usuario se le concede el acceso. Por ejemplo, en el caso del shell BASH, uno de sus archivos de arranque es el `.bashrc` (sí, se tiene un punto adelante del nombre del archivo; los nombres de archivos precedidos por puntos, también llamados archivos punto, se esconden de las listas de directorios normales). Puede concebir los scripts shell en la misma forma que en los archivos por lotes, excepto que aquellos pueden ser mucho más capaces. En particular, el script `.bashrc` tiene una naturaleza semejante al `autoexec.bat` en el mundo de Windows.

En varios paquetes de software de Linux se usan opciones específicas y que se pueden personalizar en directorios o archivos que empiezan con un `.`, en cada directorio inicial de usuario. Algunos ejemplos son `.mozilla` y `.kde`. A continuación se dan algunos archivos punto (`.`) que se encuentran presentes en el directorio inicial de cada usuario:

- ▼ `.bashrc/.profile` Archivos de configuración para BASH.
- `.tcshrc/.login` Archivos de configuración para tcsh.
- `.xinitrc` Este script anula el script predeterminado que se llama cuando usted es admitido en X Window System.
- ▲ `.Xdefaults` Este archivo contiene opciones predeterminadas que usted puede especificar para las aplicaciones en X Window System.

Cuando crea una cuenta de usuario, también se crean un conjunto de archivos punto predeterminados para ese usuario; esto es principalmente por conveniencia, para ayudar el inicio del usuario. Las herramientas de creación del usuario que se discuten más adelante le ayudan a realizar esto en forma automática. Los archivos predeterminados se almacenan bajo el directorio `/etc/skel`.

En beneficio de la uniformidad, la mayor parte de los sitios colocan los directorios iniciales en `/home` y nombran el directorio de cada usuario con el nombre de entrada de ese usuario. De este modo, si por ejemplo el nombre con el que usted fue admitido fuera "yyang", su directorio inicial sería `/home/yyang`. La excepción de esto es para algunas cuentas especiales del sistema, co-

mo la cuenta de un usuario raíz o un servicio del sistema. En Linux, suele fijarse que el directorio inicial del superusuario (del raíz) sea `/root` (pero para la mayor parte de las variantes de UNIX, como Solaris, es tradicional que el directorio inicial sea `/`). Un ejemplo de un servicio especial del sistema que podría necesitar un directorio específico de trabajo podría ser para un servidor Web cuyas páginas Web se manejan desde el directorio `/var/www/`.

En Linux, la decisión de colocar los directorios iniciales bajo `/home` es estrictamente arbitraria, pero en realidad provoca sentido de organización. De hecho, al sistema no le interesa en dónde coloquemos los directorios iniciales, en tanto que la ubicación para cada usuario se especifique en el archivo de contraseñas.

Shell (intérprete de comandos)

Cuando los usuarios son admitidos al sistema, esperan un entorno que les pueda ayudar a ser productivos. Este primer programa que los usuarios encuentran se llama *shell*. Si usted ha usado el lado del mundo de Windows, podría igualar esto con command.com, Program Manager (Administrador de programas) o Windows Explorer (Explorador de Windows) (no debe confundirse con Internet Explorer, el cual es un navegador de la Web).

Bajo UNIX/Linux, la mayor parte de los shells se basan en textos. Un shell popular predeterminado para el usuario en Linux es el Bourne Again Shell, abreviado como BASH. Linux viene con varios shells que se pueden elegir; puede ver una lista de la mayor parte de ellos en el archivo `/etc/shells`. Decidir cuál es el shell correcto para usted es, en cierto modo, como elegir una cerveza favorita; lo que es bueno para usted no lo es para todos pero, todavía, ¡todos tienden a ponerse a la defensiva respecto a su elección!

Lo que hace a Linux tan interesante es que en realidad no tiene que adherirse a la lista de shells que se suministra en `/etc/shells`. En lo más estricto de las definiciones, la entrada de la contraseña para cada usuario no le presenta una lista de cuál shell ejecutar, tanto como le presenta la lista de cuál programa ejecutar primero por el usuario. Por supuesto, la mayor parte de los usuarios prefieren que el primer programa que ejecuten sea un shell, como BASH.

El archivo `/etc/shadow`

Éste es el archivo de contraseñas cifradas. En él se almacena la información de las contraseñas cifradas para las cuentas de los usuarios. Además de la contraseña cifrada, en el archivo `/etc/shadow` también se almacena la información opcional acerca del envejecimiento de la contraseña o expiración. La introducción del archivo sombra ocurrió debido a la necesidad de separar las contraseñas cifradas del archivo `/etc/passwd`. Esto se hizo necesario en virtud de que fue creciendo la facilidad con la cual las contraseñas cifradas podían ser descifradas con el aumento en el poder de procesamiento de las computadoras de consumo (PC domésticas). La idea fue mantener el archivo `/etc/passwd` de manera que pudiera ser leído por todos los usuarios, sin almacenar las contraseñas cifradas en él, y, a continuación, hacer que el archivo `/etc/shadow` sólo pudiera ser leído por el raíz u otros programas privilegiados que requieren el acceso a esa información. Un ejemplo de ese tipo de programas sería el programa login (de concesión de acceso).

Se podría uno preguntar: “¿por qué no sólo hacer que el archivo `/etc/passwd` pudiera ser leído sólo por el raíz u otros programas privilegiados?” Bien, eso no es tan sencillo. Al tener el archivo de contraseñas abierto durante tantos años, el resto del software del sistema que creció en torno a él dependió del hecho de que ese archivo siempre podía ser leído por todos los usuarios. Cambiar esto sencillamente haría que el software fallara.

Precisamente como en el archivo **/etc/passwd**, cada línea en el **/etc/shadow** representa información acerca del usuario. Los renglones se forman por varios campos estándar, delimitándose cada uno de ellos por medio de dos puntos. Los campos son

- ▼ Nombre para obtener el acceso
- Contraseña cifrada
- Días transcurridos a partir del 1 de enero de 1970 en que la contraseña se cambió por última vez
- Días antes de los cuales la contraseña puede ser cambiada
- Días después de los cuales debe cambiarse la contraseña
- Días antes de que expire la contraseña en que debe avisarse al usuario
- Días después de que expire la contraseña en que se desactiva esa cuenta
- Días transcurridos a partir del 1 de enero de 1970 en que se desactiva esa cuenta
- ▲ Un campo reservado

Enseguida, se presenta una entrada muestra del archivo **/etc/shadow** para la cuenta **mmel** del usuario:

```
mmel:$1$HEWdPIJ.$qX/RbB.TPGcyerAVDlF4g.:12830:0:99999:7:::
```

El archivo **/etc/group**

El archivo **/etc/group** contiene una lista de los grupos, con un grupo por línea. Cada entrada de grupo en el archivo tiene cuatro campos estándar, con cada uno de ellos delimitado por dos puntos, como en los archivos **/etc/passwd** y **/etc/shadow**. Cada usuario del sistema pertenece por lo menos a un grupo, considerándose a ése como el grupo predeterminado del usuario. Entonces, si es necesario, los usuarios se pueden asignar a grupos adicionales. El lector recordará que el archivo **/etc/passwd** contiene la ID del grupo predeterminado (GID) de cada usuario. Esta GID se aplica al nombre del grupo y otros miembros del mismo en el archivo **/etc/group**. La GID debe ser única para cada grupo.

Asimismo, como en el archivo **/etc/passwd**, el mundo debe poder leer el archivo de grupos, de modo que las aplicaciones puedan probarse por asociaciones entre usuarios y grupos. Los campos de cada línea en el archivo **/etc/group** son

- ▼ **Nombre del grupo** El nombre del grupo
- **Contraseña del grupo** Ésta es opcional, pero si se fija permite que usuarios que no son parte del grupo se unan al mismo
- **ID del grupo (GID)** El equivalente numérico del nombre del grupo
- ▲ **Miembros del grupo** Una lista separada por comas

Enseguida se da una entrada muestra de un grupo en el archivo **/etc/group**:

```
bin:x:1:root,bin,daemon
```

Esta entrada es para el grupo "bin". La GID para el grupo es 1 y sus miembros son el raíz, bin y daemon.

HERRAMIENTAS PARA ADMINISTRACIÓN DE LOS USUARIOS

La parte maravillosa acerca de tener archivos de bases de datos de contraseñas que tienen un formato bien definido en texto directo es que resulta fácil para cualquiera poder escribir sus propias herramientas de administración. En efecto, muchos administradores de sitios ya han hecho esto para integrar sus herramientas junto con el resto de la infraestructura de su organización. Pueden iniciar un nuevo usuario a partir de la misma forma que les permite actualizar el teléfono corporativo y el directorio de correo electrónico, los servidores LDAP, las páginas Web, etcétera. Por supuesto, no todos desean escribir sus propias herramientas, lo cual es la razón por la que Linux viene con varias herramientas preescritas que realizan el trabajo por usted.

En esta sección, se discuten las herramientas de administración de los usuarios que se pueden usar a partir de la interfaz de línea de comandos, así como con la interfaz gráfica del usuario (GUI). Por supuesto, aprender la manera de usar las dos es la ruta preferida, ya que las dos tienen sus ventajas y su lugar.

Administración de los usuarios con la línea de comandos

Puede elegir de entre seis herramientas de línea de comandos para efectuar las mismas acciones que se realizan por medio de la herramienta GUI: **useradd**, **userdel**, **usermod**, **groupadd**, **groupdel** y **groupmod**. La ventaja poderosa de usar las herramientas de línea de comandos para la administración de usuarios, además de la rapidez, es el hecho de que, por lo común, las herramientas se pueden incorporar en otras funciones automatizadas (como los scripts).

NOTA Las distribuciones de Linux que no sean Fedora y RHEL pueden tener parámetros ligeramente diferentes de las herramientas que se usan en este texto. Para ver en qué forma su instalación particular es distinta, lea la página man del programa particular en cuestión.

useradd

Como el nombre lo implica, **useradd** le permite agregar un solo usuario al sistema. A diferencia de las herramientas GUI, esta herramienta no tiene mensajes interactivos. En lugar de ello, se deben especificar todos los parámetros en la línea de comandos.

A continuación se ilustra cómo usar esta herramienta:

```
usage: useradd [-u uid [-o]] [-g group] [-G group, ...]
                [-d home] [-s shell] [-c comment] [-m [-k template]]
                [-f inactive] [-e expire] [-p passwd] [-M] [-n] [-r] name
useradd -D [-g group] [-b base] [-s shell]
                [-f inactive] [-e expire]
```

Tome nota que cualquier cosa entre corchetes en este resumen de uso es opcional. Asimismo, ¡no se intimide por esta larga lista de opciones! Todas son bastante fáciles de usar y se describen en la tabla 4-1.

Opción	Descripción
-c comment	Permite fijar el nombre del usuario en el campo GECOS. Como con cualquier parámetro de la línea de comandos, si el valor incluye un espacio, necesitará poner el texto entre comillas. Por ejemplo, para fijar el nombre del usuario en Ying Yang, tendría que especificar -c "Ying Yang" .
-d homedir	De manera predeterminada, el directorio inicial del usuario es /home/user_name . Cuando se crea un nuevo usuario, se crea el directorio inicial de él junto con la cuenta del mismo. De modo que si quiere cambiar el valor predeterminado hacia otro lugar, puede especificar la nueva ubicación con este parámetro.
-e expire-date	Es posible que una cuenta expire después de cierta fecha. De manera predeterminada, la cuenta nunca expira. Para especificar una fecha, asegúrese de colocarla en el formato YYYY MM DD . Por ejemplo, use -e 2009 10 28 para que la cuenta expire el 28 de octubre de 2009.
-f inactive-time	Esta opción especifica el número de días después de que expira una contraseña en que la cuenta todavía se puede usar. Un valor de 0 (cero) indica que la cuenta se desactiva de inmediato. Un valor de -1 nunca permitirá que la cuenta se desactive, aun cuando la contraseña haya expirado (por ejemplo, -f 3 dejará que una cuenta exista durante tres días después de que una contraseña haya expirado). El valor predeterminado es -1 .
-g initial-group	Usando esta opción, puede especificar el grupo predeterminado que el usuario tiene en el archivo de contraseñas. Puede usar un número o nombre del grupo; sin embargo, si usa un nombre de un grupo, éste debe existir en el archivo /etc/group .
-G group[,...]	Esta opción le permite especificar grupos adicionales a los cuales pertenecerá el nuevo usuario. Si usa la opción -G , debe especificar por lo menos un grupo adicional. No obstante, puede especificar grupos adicionales separando los elementos de la lista con comas. Por ejemplo, para agregar un usuario a los grupos proyecto y admin, debe especificar -G project,admin .

Tabla 4-1. Opciones para el comando useradd

Opción	Descripción
-m [-k <i>skel-dir</i>]	De manera predeterminada, el sistema crea en forma automática el directorio inicial del usuario. Esta opción es el comando explícito para crear el directorio inicial del usuario. Parte de la creación del directorio es copiar los archivos predeterminados de configuración en él. De manera predeterminada, estos archivos vienen del directorio /etc/skel . Puede cambiar esto mediante el uso de la opción secundaria -k <i>skel-dir</i> (debe especificar -m para usar -k). Por ejemplo, para especificar el directorio /etc/adminskel , usaría -m -k /etc/adminskel .
-M	Si usa la opción -m , no puede usar -M , y viceversa. Esta opción le dice al comando que <i>no</i> cree el directorio inicial del usuario.
-n	En Linux de Red Hat se crea un nuevo grupo con el mismo nombre de la entrada del nuevo usuario, como parte del proceso de agregar a este último. Puede desactivar este comportamiento usando esta opción.
-s <i>shell</i>	Un shell de entrada del usuario es el primer programa que se ejecuta cuando un usuario entra al sistema. Éste suele ser un entorno de líneas de comandos, a menos que usted esté realizando el registro de las entradas con base en la pantalla de entrada de X Window System. De manera predeterminada, éste es el Bourne Again Shell (/bin/bash), aunque a algunos muchachos les gustan otros shells, como el Turbo C Shell (/bin/tcsh).
-u <i>uid</i>	De manera predeterminada, el programa encontrará la UID siguiente de la que se disponga y la usará. Si, por alguna razón, necesita que la UID de un nuevo usuario sea un valor particular, puede usar esta opción. Recuerde que las UID deben ser únicas para todos los usuarios.
<i>name</i>	Por último, ¡el único parámetro que <i>no</i> es opcional! Debe especificar el nombre de entrada del nuevo usuario.

Tabla 4-1. Opciones para el comando useradd (*cont.*)

usermod

El comando **usermod** le permite modificar a un usuario existente en el sistema. Funciona en gran parte de la misma manera que **useradd**. Su uso se resume a continuación:

```
usage: usermod [-u uid [-o]] [-g group] [-G group,...]
                [-d home [-m]] [-s shell] [-c comment] [-l new_name]
                [-f inactive] [-e expire] [-p passwd] [-L|-U] name
```

Cada opción que especifique al usar este comando dará por resultado que se esté modificando un parámetro en particular para el usuario. Todos los parámetros, excepto uno, cuya lista se da aquí son idénticos a los documentados para el comando **useradd**. La única excepción es **-l**.

La opción **-l** le permite cambiar el nombre de entrada del usuario. Ésta y la opción **-u** son las únicas que requieren un cuidado especial. Antes de cambiar la concesión de entrada o la UID de un usuario, debe tener la seguridad de que el usuario no ha sido admitido al sistema ni que esté ejecutando algún proceso. El cambio de esta información, si el usuario ha entrado o está ejecutando procesos, dará lugar a resultados imprevisibles.

userdel

El comando **userdel** realiza exactamente lo opuesto a **useradd**; elimina usuarios existentes. El comando directo tiene sólo un parámetro opcional y un parámetro requerido:

```
usage: userdel [-r] username
```

groupadd

Los comandos para los grupos son semejantes a los de los usuarios: sin embargo, en lugar de trabajar con usuarios por separado, se trabaja sobre grupos cuya lista aparece en el archivo **/etc/group**. Note que cambiar la información de un grupo no hace que se cambie de forma automática la información del usuario. Por ejemplo, si se elimina un grupo cuya GID es 100 y el grupo predeterminado de un usuario se especifica como 100, el grupo predeterminado de ese usuario no se actualizaría para reflejar el hecho de que el grupo ya no existe.

El comando **groupadd** agrega grupos al archivo **/etc/group**. Las opciones de la línea de comandos para este programa son como sigue:

```
usage: groupadd [-g gid [-o]] [-r] [-f] group
```

En la tabla 4-2, se describen las opciones de comandos.

groupdel

Incluso más directo que **userdel**, el comando **groupdel** elimina los grupos existentes especificados en el archivo **/etc/group**. La única información de uso que se necesita para este comando es

```
usage: groupdel group
```

en donde **group** es el nombre del grupo a eliminar.

Opción	Descripción
-g <i>gid</i>	Especifica la GID para el nuevo grupo como <i>gid</i> . Este valor debe ser único, a menos que se use la opción -o . De manera predeterminada, se elige en forma automática este valor al hallar el primer valor disponible mayor que 500 o igual a éste.
-r	De manera predeterminada, en Fedora y RHEL, búsqueda de la primera GID que sea mayor que 499. Las opciones -r le dicen a groupadd que el grupo que se está agregando es uno del sistema y debe tener la menor GID de la que se disponga por debajo de 499.
-f	Ésta es la bandera de fuerza. Ésta hará que se salga de groupadd sin error, cuando el grupo que se está a punto de agregar ya existe en el sistema. Si ése es el caso, el grupo no se alterará (o agregará de nuevo). Es una opción específica de Fedora y RHEL.
group	Se requiere esta opción. Especifica el nombre del grupo que desea agregar como <i>grupo</i> .

Tabla 4-2. Opciones para el comando **groupadd**

groupmod

El comando **groupmod** le permite modificar los parámetros de un grupo existente. Las opciones para este comando son

```
usage: groupmod [-g gid [-o]] [-n name] group
```

en donde la opción **-g** le permite cambiar la GID del grupo y la opción **-n** le permite especificar un nuevo nombre de un grupo. Por supuesto, de manera adicional, necesita especificar el nombre del grupo existente como el último parámetro.

Administradores GUI de usuarios

La ventaja obvia de utilizar la herramienta GUI es su facilidad de uso. Suele ser un asunto de apuntar y hacer clic. Muchas de las distribuciones de Linux vienen con sus propios administradores GUI de usuarios. Fedora Core viene con un programa utilitario llamado **system-config-users**, RHEL viene con un programa utilitario conocido como **redhat-config-users** y Linux

de SuSE tiene un módulo YaST que se puede llamar con **yast2 users**. Todas estas herramientas le permiten agregar/editar y mantener a los usuarios en su sistema. Estas interfaces GUI funcionan sólo bien; pero debe estar preparado para tener que cambiar en forma manual los valores establecidos para los usuarios en el caso de que no tenga acceso a las bellas entradas de GUI. La mayor parte de estas interfaces se pueden hallar en el menú System Settings (Ajustes del sistema) dentro del entorno del escritorio de GNOME o KDE. También se pueden lanzar directamente desde la línea de comandos. Para lanzar el administrador GUI de usuarios de Fedora, teclearía

```
[root@serverA ~]# system-config-users
```

Aparecerá una ventana semejante a la de la figura 4-2.

En SuSE, para lanzar el módulo YaST de administración de los usuarios, teclearía

```
serverA:~ # yast2 users
```

Aparecerá una ventana semejante a la de la figura 4-3.

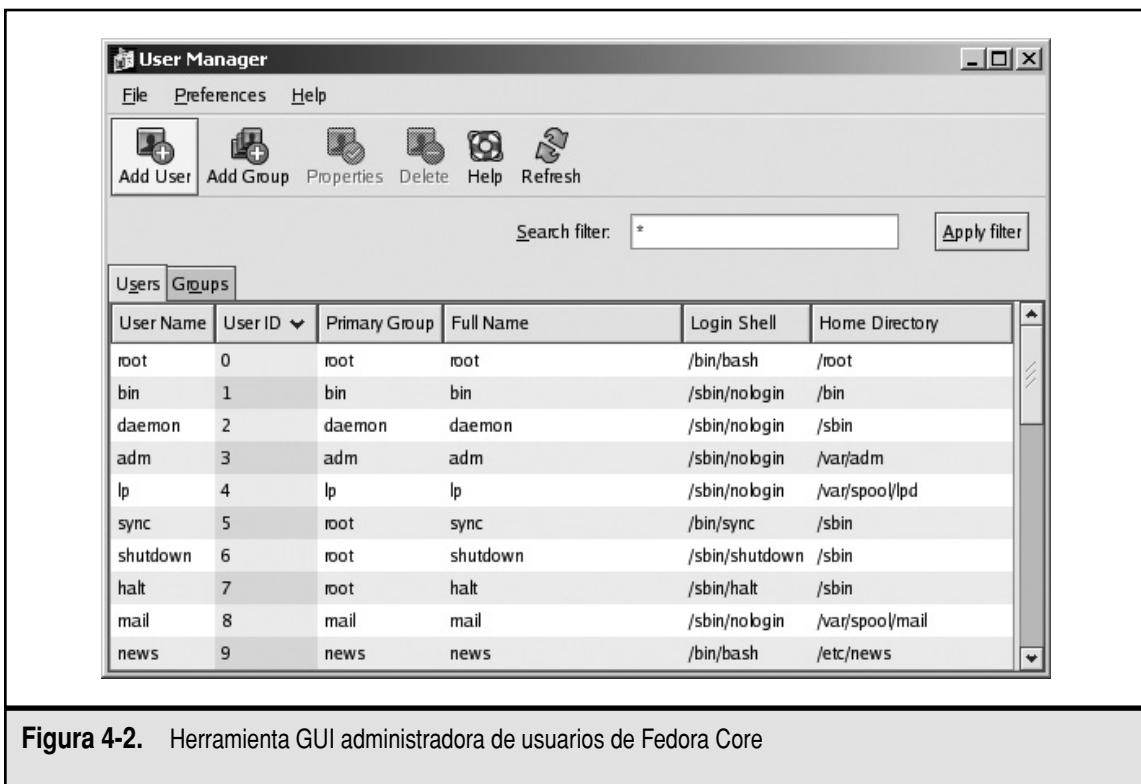


Figura 4-2. Herramienta GUI administradora de usuarios de Fedora Core

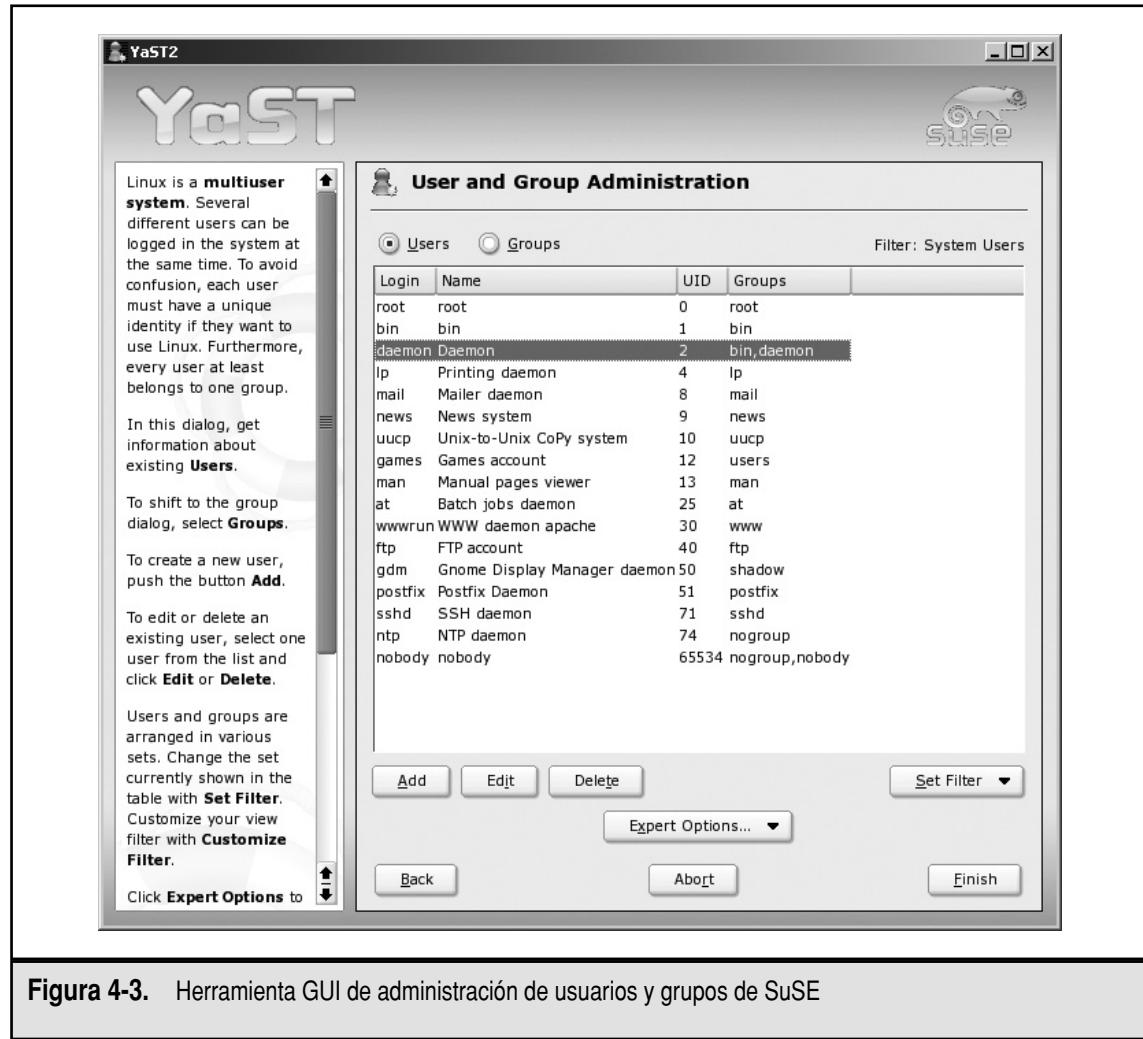


Figura 4-3. Herramienta GUI de administración de usuarios y grupos de SuSE

USUARIOS Y LOS PERMISOS DE ACCESO

Linux determina si un usuario o grupo tienen acceso o no a los archivos, programas u otros recursos de un sistema al revisar los permisos totales efectivos en el recurso. El modelo tradicional de permisos en Linux es muy sencillo; se basa en cuatro tipos o reglas de acceso. Los tipos posibles de acceso son

- ▼ (r) Permiso de leer
- (w) Permiso de escribir

- (x) Permiso de ejecutar
- ▲ (-) Permiso o acceso negado

Además, estos permisos se pueden aplicar a tres clases de usuarios. Las clases son

- ▼ **Propietario** El propietario del archivo o aplicación
- **Grupo** El grupo que posee el archivo o aplicación
- ▲ **Todos** Todos los usuarios

Los elementos de este modelo se pueden combinar de varias maneras, para permitir o negar el acceso a un usuario (o grupo) a cualquier recurso del sistema. Sin embargo, en Linux, hay necesidad de un tipo adicional de mecanismo de concesión de permisos. Esta necesidad se presenta porque toda aplicación en Linux debe ejecutarse en el contexto de un usuario. Esto se explica en la sección siguiente, sobre los programas SetUID y SetGID.

Comprendiendo los programas SetUID y SetGID

Normalmente, cuando un usuario ejecuta un programa, hereda todos los derechos (o la falta de ellos) que el usuario tiene. Si el usuario no puede leer el archivo `/var/log/messages`, tampoco lo puede hacer el programa. Note que este permiso puede ser diferente de los permisos del usuario a quien pertenece el programa (por lo común llamado *el binario*). Por ejemplo, al usuario raíz le pertenece el programa `ls` (el cual se usa para generar listas de directorios). Sus permisos se fijan de modo que todos los usuarios del sistema puedan ejecutar el programa. De donde, si el usuario yyang ejecuta `ls`, ese caso de `ls` queda ligado por los permisos concedidos al usuario yyang, no al raíz.

Sin embargo, existe una excepción. Los programas se pueden etiquetar con lo que se conoce como *bit de SetUID*, lo cual permite que un programa se ejecute con los permisos del propietario de ese programa, no con los del usuario que lo está ejecutando. Usando una vez más `ls` como ejemplo, al fijar el bit de SetUID en él y tener al raíz como propietario del archivo significa que si el usuario yyang ejecuta `ls`, ese caso de `ls` se ejecutará con los permisos del raíz, no con los permisos de yyang. El *bit de SetGID* funciona de la misma manera, excepto que en lugar de aplicarlo al propietario del archivo, se aplica a los valores del grupo del archivo.

Para activar el bit de SetUID o el de SetGID, necesita usar el comando `chmod`. Para obtener un programa SetUID, coloque el prefijo 4 a cualquiera que sea el valor del permiso que esté a punto de asignarle. Para obtener un programa SetGID, coloque el prefijo 2 a cualquiera que sea el permiso que esté a punto de asignarle. Por ejemplo, para hacer que `/bin/ls` sea un programa SetUID (lo cual, a propósito, es una mala idea), usaría este comando:

```
[root@serverA /root]# chmod 4755 /bin/ls
```

PLUGGABLE AUTHENTICATION MODULES (PAM)

Los Pluggable Authentication Modules (PAM) (Módulos enchufables de autenticación) permiten el uso de un mecanismo centralizado de autenticación en los sistemas Linux/UNIX. Además de suministrar un sistema común de autenticación en un sistema, el uso de los PAM proporciona una gran flexibilidad y control sobre la autenticación para los desarrolladores de aplicaciones así como para los desarrolladores de sistemas.

De modo tradicional, los programas que conceden el acceso a los usuarios a los recursos del sistema realizaban la autenticación del usuario a través de algún mecanismo integrado. Aun cuando esto funcionó de maravilla durante mucho tiempo, no era fácil llevar a una escala mayor el procedimiento y se requirieron métodos más elaborados. Esto condujo a que varios hackers muy desagradables se abstrajeran en el mecanismo de autenticación. Guiándose por Solaris, los muchachos de Linux crearon su propia implementación de los Pluggable Authentication Modules (PAM).

La idea que se encuentra detrás de PAM es que, en lugar de que las aplicaciones leyeren el archivo de contraseñas, sencillamente le pedirían a PAM realizar la autenticación. Entonces PAM podría usar cualquiera que fuera el mecanismo de autenticación que deseara el administrador del sistema. Para muchos sitios, el mecanismo que seleccionan todavía es un simple archivo de contraseñas. ¿Y por qué no? Hace lo que queremos. La mayor parte de los usuarios entienden la necesidad de ello y es un método bien probado para lograr que se haga el trabajo.

En esta sección discutimos el uso de PAM bajo la versión Fedora Core de Linux. Se debe hacer notar que aun cuando la colocación de los archivos puede no ser exactamente la misma en otras distribuciones, los archivos de configuración y los conceptos subyacentes todavía se aplican.

Cómo funciona PAM

PAM es para otros programas lo que una DLL es para una aplicación de Windows: es sencillamente una biblioteca. Cuando los programas necesitan realizar la autenticación de alguien, llaman una función que existe en la biblioteca PAM. Ésta proporciona una biblioteca de funciones que una aplicación puede usar para pedir que se autentique un usuario.

Cuando se llama, PAM revisa el archivo de configuración para esa aplicación. Si no existe un archivo de configuración, usa un archivo predeterminado de ese tipo. Este archivo de configuración le dice a la biblioteca qué tipos de comprobaciones es necesario que se hagan para autenticar al usuario. Con base en esto, se llama al módulo apropiado (los muchachos de Fedora y RHEL pueden ver estos módulos en el directorio `/lib/security`).

Este módulo puede comprobar cualquier número de cosas. Tan sólo puede revisar el archivo `/etc/passwd` o el `/etc/shadow`, o bien, puede realizar una comprobación más completa como llamar a un servidor LDAP.

NOTA El sitio Web de PAM (<http://www.kernel.org/pub/linux/libs/pam/>) ofrece una lista completa de los módulos de los que se dispone.

Una vez que el módulo ha hecho la determinación, se regresa un mensaje de “autenticado/no autenticado” a la aplicación que llama.

Si esto se siente como una gran cantidad de pasos para lo que debe ser una simple comprobación, está usted casi en lo correcto. Aun cuando se siente como una gran cantidad de pasos, cada módulo que aquí interviene es muy pequeño y realiza su trabajo con gran rapidez. Desde el punto de vista de un usuario, no debe ser notable la diferencia de rendimiento entre una aplicación en la que se usa PAM y otra que no lo haga. Desde el punto de vista de un administrador de sistema y de un desarrollador, la flexibilidad que ofrece este sistema es increíble y resulta una adición muy bienvenida.

Archivos de PAM y sus ubicaciones

En un sistema del tipo de Fedora, PAM pone sus archivos de configuración en ciertos lugares. En la tabla 4-3, se da una lista de las ubicaciones de estos archivos y sus definiciones.

Ubicación del archivo	Definición
/lib/security	Módulos de autenticación dinámicamente cargados a los que llama la biblioteca PAM actual.
/etc/security	Archivos de configuración para los módulos ubicados en /lib/security.
/etc/pam.d	Archivos de configuración para cada aplicación que usa PAM. Si una aplicación que usa PAM no tiene un archivo específico de configuración, el archivo predefinido se usa de manera automática.

Tabla 4-3. Directorios importantes de PAM

Si se mira la lista de las ubicaciones de los archivos en la tabla 4-3, tiene uno que preguntarse por qué PAM necesita tantos archivos diferentes de configuración. “¿Un archivo de configuración por aplicación? ¡Eso parece una locura!” Bien, puede ser que no. La razón por la que PAM permite esto es que no todas las aplicaciones se crearon igual. Por ejemplo, un servidor de correo POP en el que se usa el servidor de correo Qpopper puede querer que se permita a todos los usuarios de un sitio recoger el correo, pero el programa para entrar puede querer que sólo se permita que ciertos usuarios puedan entrar a la consola. Con el fin de hacer frente a esto, PAM necesita un archivo de configuración para correo POP que es diferente a la configuración para el programa de entrada.

Configuración de PAM

Los archivos de configuración que se discutirán aquí son los ubicados en el directorio /etc/pam.d. Si desea cambiar los archivos de configuración que se aplican a módulos específicos, del directorio /etc/security, debe consultar la documentación que viene con el módulo (recuerde, PAM es sólo un marco de referencia). Los módulos específicos pueden ser escritos por cualquiera).

La naturaleza de un archivo de configuración de PAM es muy interesante debido a su calidad de “apilables”. Es decir, cada línea de un archivo de configuración se evalúa en el transcurso del proceso de autenticación (con las excepciones que se muestran enseguida). Cada línea especifica un módulo que realiza una tarea de autenticación y responde con una bandera de éxito o de falla. Al programa de aplicación que llama a PAM se le pasa de regreso un resumen de los resultados.

NOTA Por “falla”, no queremos dar a entender que el programa no funcionó. Más bien, queremos decir que cuando se realizó un proceso para verificar si un usuario podía hacer algo, el valor de respuesta fue “NO”. En PAM se usan los términos “éxito” y “falla” para representar esta información que se pasa de regreso a la aplicación que llama.

Cada archivo consta de líneas con el formato siguiente:

```
module_type    control_flag    module_path    arguments
```

en donde **module_type** representa uno de cuatro tipos de módulos: **auth**, **account**, **session** o **password**. Los comentarios se deben comenzar con el carácter de número (#). En la tabla 4-4, se da una lista de estos tipos de módulos y sus funciones.

El **control_flag** nos permite especificar la manera en que queremos tratar con el éxito o la falla de un módulo en particular de autenticación. En la tabla 4-5 se describen las banderas de control.

El **module_path** especifica la trayectoria actual del directorio del módulo que realiza la tarea de autenticación. Los módulos suelen almacenarse bajo el directorio /lib/security. Para obtener una lista completa de los módulos, visite el sitio Web de PAM (<http://www.kernel.org/pub/linux/libs/pam>).

La entrada final en una línea de configuración de PAM es **arguments**. Éstos son los parámetros que se pasan al módulo de autenticación. Aunque los parámetros son específicos para cada módulo, se tienen algunas opciones genéricas que se pueden aplicar a todos los módulos. En la tabla 4-6 se describen estos argumentos.

Tipo de módulo	Función
auth	Instruye al programa de aplicación que pida al usuario una contraseña y, a continuación, concede los privilegios tanto de los usuarios como de los grupos.
account	No realiza la autenticación pero determina el acceso a partir de otros factores, como el momento del día o la ubicación del usuario. Por ejemplo, a la entrada del raíz sólo se le puede dar acceso por la consola de esta manera.
session	Especifica cuáles acciones, si las hay, es necesario realizar antes o después de que se dé acceso a un usuario (por ejemplo, registrar la conexión).
password	Especifica el módulo que permite a los usuarios cambiar su contraseña (si es apropiado).

Tabla 4-4. Tipos de módulos PAM

Bandera de control	Descripción
required	Si se especifica esta bandera, el módulo <i>debe</i> tener éxito en la autenticación del individuo. Si falla, el valor de resumen que se regrese debe ser de falla.
requisite	Esta bandera es semejante a required ; sin embargo, si requisite falla la autenticación, no se llama a los módulos que se encuentran después de ella en el archivo de configuración y, de inmediato, se regresa una falla a la aplicación. Esto nos permite requerir que sean verdaderas ciertas condiciones, incluso antes de aceptar un intento de entrada (por ejemplo, el usuario está en la red de área local y no puede provenir de Internet).
sufficient	Si un módulo sufficient regresa un éxito y no hay más banderas de control required o sufficient en el archivo de configuración, PAM responde con un éxito a la aplicación que llama.
optional	Esta bandera le permite a PAM continuar verificando otros módulos, incluso si ésta ha fallado. Usted querrá usar esto cuando se le permite la entrada al usuario, incluso si ha fallado un módulo en particular.

Tabla 4-5. Banderas de control de PAM

Un archivo ejemplo de configuración de PAM

Examinemos un archivo muestra de configuración de PAM, /etc/pam.d/login:

```
 #%PAM-1.0
auth    required    pam_securetty.so
auth    required    pam_stack.so service=system-auth
auth    required    pam_nologin.so
account  required   pam_stack.so service=system-auth
password required   pam_stack.so service=system-auth
# pam_selinux.so close should be the first session rule
session  required   pam_selinux.so close
session  required   pam_stack.so service=system-auth
session  optional   pam_console.so
# pam_selinux.so open should be the last session rule
session  required   pam_selinux.so multiple open
```

Argumento	Descripción
<code>Debug</code>	Envía información de depuración a los registros lógicos del sistema.
<code>no_warn</code>	No le da mensajes de advertencia a la aplicación que llama.
<code>use_first_pass</code>	No le pide por segunda vez al usuario una contraseña. En lugar de ello, la contraseña que se introdujo en el módulo auth precedente debe volverse a usar para la autenticación del usuario. (Esta opción sólo es para los módulos auth y password .)
<code>try_first_pass</code>	Esta opción es semejante a <code>use_first_pass</code> , en donde no se le pide por segunda vez al usuario una contraseña. Sin embargo, si la contraseña existente hace que el módulo responda con una falla, entonces se pide al usuario de nuevo una contraseña.
<code>use_mapped_pass</code>	Este argumento instruye al módulo que tome la señal de autenticación de texto claro que introdujo un módulo anterior y lo use para generar una clave de cifrado/ descifrado con la cual almacenar o recuperar con seguridad la señal de autenticación requerida por este módulo.
<code>Expose_account</code>	Este argumento permite a un módulo ser menos discreto acerca de la información de la cuenta; según el ajuste estimado por el administrador del sistema.

Tabla 4-6. Argumentos de configuración de PAM

Podemos ver que la primera línea empieza con un símbolo de número y, por lo tanto, es un comentario. Por consiguiente, podemos ignorarlo. Vayamos a la línea 2:

```
auth      required      pam_securetty.so
```

Dado que el **module_type** es **auth**, PAM querrá una contraseña. La **control_flag** se fija como **required**, de modo que este módulo debe responder con un éxito o la entrada fallará. El propio módulo, **pam_securetty.so**, verifica que sólo se tengan entradas a la cuenta raíz en las terminales mencionadas en el archivo **/etc/securetty**. No hay argumentos en esta línea.

```
auth      required      pam_stack.so service=system-auth
```

De modo semejante a la primera línea **auth**, en la línea 3 se necesita una contraseña para la autenticación, y si la contraseña falla, el proceso de autenticación dará como respuesta una bandera de falla a la aplicación que llama. El módulo **pam_stack.so** permite llamar desde el interior de la pila para obtener un servicio en particular, o bien, la pila definida para otro servicio. El argumento **service=system-auth** en este caso le dice a **pam_stack.so** que ejecute la pila definida para el servicio system-auth (system-auth también es otra configuración de PAM bajo el directorio **/etc/pam.d**).

```
auth      required      pam_nologin.so
```

En la línea 4, el módulo **pam_nologin.so** verifica la existencia del archivo **/etc/nologin**. Si se encuentra, sólo se permite entrar al raíz; a los otros no se les admite con un mensaje de error. Si el archivo no existe, siempre responde con un éxito.

```
account    required      pam_stack.so service=system-auth
```

En la línea 5, como el **module_type** es **account**, el módulo **pam_stack.so** actúa de manera diferente. En forma silenciosa, verifica si al usuario todavía se le permite entrar (por ejemplo, “¿ha expirado su contraseña?”). Si la comprobación de todos los parámetros se realiza en forma satisfactoria, responderá con un éxito.

Los mismos conceptos se aplican al resto de las líneas del archivo **/etc/pam.d/login** (así como a otros archivos de configuración bajo el directorio **/etc/pam.d**).

Si necesita más información respecto a lo que realiza un módulo PAM en particular o acerca de los argumentos que acepta, puede consultar la página man del módulo. Por ejemplo, para averiguar más en relación con el módulo **pam_sselinux.so**, emitiría el comando

```
[root@serverA ~]# man pam_sselinux
```

El archivo “other”

Como mencionamos con anterioridad, si PAM no puede hallar un archivo de configuración que sea específico para una aplicación, en lugar de él usará un archivo genérico de configuración. Este archivo se llama **/etc/pam.d/other**. De manera predeterminada, al archivo de configuración “other” se le da un valor paranoide de manera que se registren todos los intentos de autenticación y, de inmediato, se nieguen. Se recomienda que lo conserve de esa manera.

“¡DOH! ¡No puedo entrar!”

No se preocupe; estropear un ajuste en un archivo PAM de configuración le sucede a cualquiera. Considérelo como parte de ponerse al tanto. Lo primero que debe hacer: no entrar en pánico. Como la mayor parte de los errores de configuración bajo Linux, puede arreglar las cosas inicializando en el modo de un solo usuario (vea el capítulo 7) y arreglando el archivo con error.

Si ha estropeado su archivo de configuración de entrada y necesita regresarlo a un estado sano, a continuación tiene un ajuste seguro que puede meter:

```
auth      required      pam_unix.so
account    required      pam_unix.so
password  required      pam_unix.so
session   required      pam_unix.so
```

Este ajuste dará a Linux el comportamiento predeterminado de mirar sencillamente en el archivo `/etc/passwd` o en el `/etc/shadow` en busca de una contraseña. Esto debe ser suficientemente bueno como para hacer que usted regrese, ¡en donde pueda hacer los cambios que quiera hacer!

NOTA El módulo `pam_unix.so` es el que facilita este comportamiento. Es el módulo estándar de autenticación de UNIX. Según la página man del módulo, utiliza llamadas estándar desde las bibliotecas del sistema para recuperar y fijar la información de la cuenta así como su autenticación. Por lo común, esto se obtiene del archivo `/etc/passwd` y también del `/etc/shadow`, si éste se encuentra activado.

Depuración de PAM

Como muchos otros servicios de Linux, PAM hace un uso excelente de los archivos de registro del sistema (puede leer más acerca de ellos en el capítulo 8). Si las cosas no funcionan de la manera en que usted desea que lo hagan, empiece por mirar en el extremo final de estos archivos y vea si PAM está comprendiendo lo que sucedió. Más que probable, lo está. Entonces debe usted poder usar esta información para cambiar sus ajustes y resolver su problema. El archivo principal de registro del sistema que hay que monitorear es `/var/log/messages`.

UN GRAN RECORRIDO

La mejor manera de ver interactuar entre sí muchas de las herramientas utilitarias discutidas en este capítulo es mostrarlas en funcionamiento. En esta sección, realizaremos en procedimiento paso a paso para crear, modificar y eliminar usuarios y grupos. También se presentan y usan algunos comandos nuevos que no se mencionaron pero que también son útiles y pertinentes para la administración de los usuarios de un sistema.

Creación de usuarios con useradd

Agregue cuentas nuevas de usuarios y asigne contraseñas con los comandos `useradd` y `passwd`,

1. Cree un nuevo usuario cuyo nombre completo sea “Ying Yang,” con el nombre de acceso (nombre de la cuenta) de `yyang`. Teclee

```
[root@serverA ~]# useradd -c "Ying Yang" yyang
```

Este comando creará una nueva cuenta de usuario llamada `yyang`. Se creará el usuario con los atributos predeterminados usuales de Fedora Core. La entrada en el archivo `/etc/passwd` será

```
yyang:x:500:500:Ying Yang:/home/yyang:/bin/bash
```

Partiendo de esta entrada, podrá hacer estas afirmaciones referentes a los valores predeterminados del nuevo usuario, de Fedora Core (y RHEL):

- ▼ El número UID es el mismo que el número GID.
 - El shell predeterminado para los nuevos usuarios es el bash (/bin/bash).
 - ▲ Se crea en forma automática un directorio inicial para todos los usuarios nuevos (por ejemplo, /home/yyang).
2. Use el comando **passwd** para crear una contraseña nueva para el nombre de usuario yyang. Fije la contraseña como **19ang19** y repita la misma contraseña cuando se le pida. Teclee

```
[root@serverA ~]# passwd yyang
Changing password for user yyang.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

3. Cree otra cuenta de usuario llamada **mmellow** para el usuario cuyo nombre completo es "Mel Mellow", pero en esta ocasión cambie el comportamiento predeterminado de Fedora de crear un grupo que tenga el mismo nombre que el del usuario (es decir, en lugar de lo anterior, este usuario pertenecerá al grupo general **users**). Teclee

```
[root@serverA ~]# useradd -c "Mel Mellow" -n mmellow
```

4. Use el comando **id** para examinar las propiedades del usuario mmellow. Teclee

```
[root@serverA ~]# id mmellow
uid=501(mmellow) gid=100(users) groups=100(users)
```

5. Una vez más, use el comando **passwd** para crear una contraseña nueva para la cuenta mmellow. Fije la contraseña como **2owl78** y repita la misma contraseña cuando se le pida. Teclee

```
[root@serverA ~]# passwd mmellow
```

6. Cree la cuenta del usuario final llamada **bogususer**. Pero en esta ocasión, especifique que el shell del usuario sea el tcsh y haga que el grupo primario predeterminado del usuario sea el grupo "games" del sistema. Teclee

```
[root@serverA ~]# useradd -s /bin/tcsh -g games bogususer
```

7. Examine el archivo **/etc/passwd** en busca de la entrada del usuario bogususer. Teclee

```
[root@serverA ~]# grep bogususer /etc/passwd
bogususer:x:502:20::/home/bogususer:/bin/tcsh
```

Con base en esta entrada, puede ver que

- ▼ La UID es 502.
- La GID es 20.
- También se crea un directorio inicial para el usuario bajo el directorio **/home**.
- ▲ El shell del usuario es /bin/tcsh.

Creación de grupos con groupadd

A continuación, cree un par de grupos, no del sistema y del sistema.

1. Cree un nuevo grupo llamado **research**. Teclee

```
[root@serverA ~]# groupadd research
```

2. Examine la entrada del grupo research en el archivo **/etc/group**. Teclee

```
[root@serverA ~]# grep research /etc/group  
research:x:501:
```

Esta salida muestra que la ID del grupo para el research es 501.

3. Cree otro grupo llamado **sales**. Teclee

```
[root@serverA ~]# groupadd sales
```

4. Cree el grupo final llamado **bogus** y, además, fuerce a este grupo a que sea uno del sistema (es decir, la GID será menor que 499). Teclee

```
[root@serverA ~]# groupadd -r bogus
```

5. Examine la entrada del grupo bogus en el archivo **/etc/group**. Teclee

```
[root@serverA ~]# grep bogus /etc/group  
bogus:x:101:
```

Esta salida muestra que la ID del grupo para el bogus es 101.

Modificación de los atributos de los usuarios con usermod

Ahora intente usar el comando **usermod** para cambiar las ID del usuario y del grupo de un par de cuentas.

1. Use el comando **usermod** para cambiar la ID del usuario (UID) de la bogususer a 600. Teclee

```
[root@serverA ~]# usermod -u 600 bogususer
```

2. Use el comando **id** para ver sus cambios. Teclee

```
[root@serverA ~]# id bogususer  
uid=600(bogususer) gid=20(games) groups=20(games)
```

La salida muestra la nueva UID (600) del usuario.

3. Use el comando **usermod** para cambiar la ID del grupo primario (GID) de la cuenta bogususer a la del grupo bogus (GID = 101) y también fije una fecha de expiración para la cuenta del 12-12-2009. Teclee

```
[root@serverA ~]# usermod -g 101 -e 2009-12-12 bogususer
```

4. Vea sus cambios con el comando **id**. Teclee

```
[root@serverA ~]# id bogususer  
uid=600(bogususer) gid=101(bogus) groups=101(bogus)
```

5. Use el comando **chage** para ver la nueva información de expiración de la cuenta, para el usuario. Teclee

```
[root@serverA ~]# chage -l bogususer
Minimum:          0
Maximum:         99999
Warning:           7
Inactive:        -1
Last Change:      Feb 18, 2007
Password Expires: Never
Password Inactive: Never
Account Expires:   Dec 12, 2009
```

Modificación de los atributos de los grupos con groupmod

Ahora trate de usar el comando **groupmod**.

1. Use el comando **groupmod** para renombrar el grupo bogus como **bogusgroup**. Teclee

```
[root@serverA ~]# groupmod -n bogusgroup bogus
```

2. Use una vez más el comando **groupmod** para cambiar la ID del grupo (GID) del bogusgroup a 600. Teclee

```
[root@serverA ~]# groupmod -g 600 bogusgroup
```

3. Vea sus cambios para el bogusgroup en el archivo **/etc/group**. Teclee

```
[root@serverA ~]# grep bogusgroup /etc/group
```

Eliminación de grupos y usuarios con groupdel y userdel

Intente usar los comandos **groupdel** y **userdel** para eliminar grupo y usuarios, respectivamente.

1. Use el comando **groupdel** para borrar el grupo bogusgroup. Teclee

```
[root@serverA ~]# groupdel bogusgroup
```

Notará que, en consecuencia, se eliminará la entrada del grupo bogusgroup en el archivo **/etc/group**.

2. Use el comando **userdel** para borrar el usuario bogususer que creó con anterioridad. En el mensaje del shell, teclee

```
[root@serverA ~]# userdel -r bogususer
```

NOTA Cuando ejecuta el comando **userdel** sólo especificando el nombre de concesión de entrada del usuario en la línea de comandos (por ejemplo, **userdel bogususer**), se eliminarán en forma automática todas las entradas en los archivos **/etc/passwd** y **/etc/shadow**, así como las referencias en el archivo **/etc/group**. Pero si usa el parámetro opcional **-r** (por ejemplo, **userdel -r bogususer**), también se eliminarán todos los archivos que pertenezcan al usuario en el directorio inicial de ese usuario.

RESUMEN

En este capítulo se documenta la naturaleza de los usuarios bajo Linux. Gran parte de lo que lea aquí también se aplica a otras variantes de UNIX, lo cual facilita más la administración de los usuarios en entornos heterogéneos con los diferentes *NIX.

Éstos son los temas más significativos cubiertos en este capítulo:

- ▼ Cada usuario recibe una UID única.
- Cada grupo recibe una GID única.
- El archivo **/etc/passwd** aplica las UID a los nombres de los usuarios.
- Linux maneja contraseñas cifradas de múltiples maneras.
- Linux incluye herramientas que le ayudan a administrar a los usuarios.
- Si decidiera escribir sus propias herramientas para administrar las bases de datos de los usuarios, ahora entenderá el formato para hacerlo.
- ▲ PAM, Pluggable Authentication Modules, es una manera genérica de Linux de manejar múltiples mecanismos de autenticación.

Estos cambios son bastante significativos para un administrador que viene del entorno de Windows XP/NT/200x y puede resultar un poco difícil al principio. Sin embargo, no se preocupe; el modelo de seguridad de Linux/UNIX es bastante directo, de modo que debe sentirse cómodo con la manera en que todo funciona.

Si le atrae la idea de ponerse a estructurar sus propias herramientas para administrar a los usuarios, definitivamente consulte libros sobre el lenguaje de programación Perl. Éste es notablemente bien apropiado para manipular datos tabulares (como el archivo **/etc/passwd**). Con las facilidades de trabajo en red de Perl y el soporte de MS Windows, Linux incluso le permite estructurar una herramienta **adduser** de plataforma cruzada que puede crear y establecer tanto cuentas de Linux como de MS Windows. Con tantos libros sobre Perl que hay por allí, cada uno con un ángulo un poco diferente y suponiendo un nivel también un poco distinto de fundamentos de programación, resulta difícil recomendar un solo libro. Tómese algo de tiempo y hojee unos cuantos en su librería próxima.

CAPÍTULO 5



La línea de comandos

El nivel de poder, control y flexibilidad que la línea de comandos ofrece a los usuarios de UNIX/Linux ha sido una de sus cualidades más atractivas y más duraderas. Aunque, también existe un lado petulante de esto: para los iniciados, la línea de comandos también puede producir emociones extremas, incluyendo admiración temerosa, frustración e irritación. Los observadores casuales de los gurús de UNIX a menudo se quedan atónitos de los resultados que se obtienen al introducir cuidadosamente unos cuantos comandos. Por fortuna, este poder hace que UNIX sea menos intuitivo para el usuario promedio. Por esta razón, se han escrito unas cuantas entradas con interfaz gráfica del usuario (GUI) para varias herramientas, funciones y utilidades de UNIX / Linux.

Sin embargo, los usuarios más experimentados, encuentran que es difícil para una GUI presentar todas las opciones de las que se dispone. Lo típico sería que al hacerlo así sólo daría como resultado algo tan complicado como el equivalente en líneas de comandos. Después de que todo se ha dicho y realizado, el hecho sigue siendo que en verdad se observa *excelente* hacer cosas en la línea de comandos.

Antes de empezar nuestro estudio de la interfaz de líneas de comandos bajo Linux, comprenda que este capítulo está lejos de ser un recurso exhaustivo. En lugar de tratar de cubrir todas las herramientas sin profundidad, hemos preferido describir por completo un puñado de ellas que creemos que son las más críticas para el trabajo cotidiano.

NOTA Para este capítulo, suponemos que ha entrado al sistema como un usuario normal y que el X Window System está montado y ejecutándose en ese sistema. Por ejemplo, si está usando el entorno GNOME de escritorio, puede iniciar una terminal virtual en la cual emita los comandos. Al hacer clic derecho sobre el escritorio se debe presentar un menú que permitirá lanzar una terminal virtual. El menú sensible al contexto puede tener una opción en el mismo en la que se lea algo como "Open Terminal" (Abrir terminal) o "Launch Terminal" (Lanzar terminal). Si no tiene esa opción particular, busque una opción en el menú que diga Run Command (Ejecutar comando). Después de que aparezca el cuadro de diálogo Run, entonces puede teclear el nombre de un emulador de terminal (por ejemplo, xterm, gnome-terminal o konsole) en ese cuadro de texto. Todos los comandos que introduzca en este capítulo deben teclearse en la ventana de la terminal virtual.

UNA INTRODUCCIÓN A BASH

En el capítulo 4, aprendió que uno de los parámetros para la entrada de contraseña de un usuario es la del shell (intérprete de comandos) de la concesión de acceso de ese usuario, el cual es el primer programa que se ejecuta cuando un usuario entra a una estación de trabajo. El shell es comparable al Program Manager de Windows, excepto que, por supuesto, el programa de shell que se use es arbitrario.

La definición formal de un shell es: un intérprete del lenguaje de comandos que ejecuta estos últimos. Una definición menos formal podría ser: sencillamente un programa que proporciona una interfaz para el sistema. En particular, el Bourne Again Shell (BASH) es una interfaz sólo para líneas de comandos que contiene un puñado de comandos integrados, la capacidad de lanzar otros programas y la capacidad para controlar programas que se han lanzado desde él (control de tareas). En principio, podría parecer simple, pero el lector empezará a darse cuenta que el shell es una herramienta muy poderosa.

Existen diversos shells, la mayor parte con características semejantes pero con diferentes medios de implementarlas. Una vez más, para fines de comparación, puede concebir los diferentes shells como si fueran semejantes a los navegadores de la Web; entre los diversos navegadores, la

funcionalidad básica es la misma: presentar el contenido de la Web. En cualquier situación como ésta, todos proclaman que su shell es mejor que los otros, pero en realidad todo se reduce a una cuestión de preferencia personal.

En esta sección, examinaremos algunos de los comandos integrados de BASH. Una referencia completa sobre BASH podría ser con facilidad un libro por sí mismo, de modo que nos apegaremos a los comandos que más afectan las operaciones diarias de un administrador de sistema. Sin embargo, se recomienda con amplitud que llegue a estudiar otras funciones y operaciones de BASH. No hay escasez de libros excelentes sobre el tema. Conforme se acostumbre a BASH, puede seleccionar con facilidad otros shells. Si está administrando un sitio grande, con muchos usuarios, le resultará ventajoso familiarizarse con tantos shells como sea posible. En realidad es bastante fácil seleccionar otro shell, ya que las diferencias entre ellos son sutiles.

Control de tareas

Al trabajar en el entorno de BASH, puede arrancar múltiples programas a partir del mismo mensaje. Cada programa es una tarea. Siempre que se inicia una tarea, absorbe la terminal (éste es un retroceso a los días cuando se usaron terminales en realidad no inteligentes, como las VT100 y las Wyse 50, como interfaces para la máquina). En las máquinas de la actualidad, la terminal es la interfaz de texto directo que usted ve cuando inicializa la máquina o la ventana creada por el X Window System, en las cuales se ejecuta BASH (en el X Window System, las interfaces de terminal se llaman pseudo-tty o, abreviado, pty). Si una tarea tiene el control de la terminal, puede emitir códigos de control de modo que las interfaces de sólo texto (por ejemplo, el lector de correo Pine) se puedan hacer más atractivas. Una vez que se termina el programa, le regresa el control completo a BASH y se vuelve a presentar un mensaje para el usuario.

Sin embargo, no todos los programas requieren esta clase de control de la terminal. A algunos, incluyendo programas que proporcionan interfaz con el usuario a través del X Window System, se les puede dar instrucciones para dejar de controlar la terminal y permitir que BASH presente un mensaje al usuario, aun cuando todavía se esté ejecutando el programa que se haya llamado.

En el ejemplo que sigue, teniendo al usuario yyang admitido en el sistema, ese usuario lanza el navegador Firefox de la Web, con la condición adicional de que el programa (Firefox) deje de controlar la terminal (esta condición se representa por el sufijo ampersand):

```
[yyang@serverA ~]$ firefox &
```

De inmediato que oprima ENTER, BASH presentará de nuevo su mensaje. A esto se le conoce como pasar a segundo plano la tarea. Las personas que recuerden Windows NT antes de la versión 4, recordarán que se tenía algo semejante con el comando Start.

Si un programa ya se está ejecutando y tiene el control de la terminal, usted puede hacer que deje de realizar ese control al oprimir CTRL-Z en la ventana de la terminal. Esto detendrá la tarea (o programa) en ejecución y regresará el control a BASH, de modo que usted pueda introducir nuevos comandos.

En cualquier momento, puede averiguar a cuántas tareas BASH le está siguiendo el rastro, al teclear este comando:

```
[yyang@serverA ~]$ jobs  
[1] + Running firefox &
```

Los programas en ejecución, cuya lista se dé, se encontrarán en uno de dos estados: en ejecución o detenidos. La salida del ejemplo que se acaba de dar hace ver que el programa Firefox está en ejecución. La salida muestra también el número de tareas, en la primera columna: [1].

Para regresar una tarea al primer plano; es decir, para regresarle el control de la terminal, usaría el comando **fg** (foreground, llevar al primer plano), como éste:

```
[yyang@serverA ~]$ fg number
```

en donde **number** es el número de la tarea que quiere en el primer plano. Por ejemplo, para colocar en el primer plano el programa Firefox (que tiene el número de tarea 1) que se lanzó al principio, teclee

```
[yyang@serverA ~]$ fg 1  
firefox
```

Si se detiene una tarea (es decir, si queda en estado de detenida), puede hacer que se ejecute de nuevo en el segundo plano, con lo cual se permite que conserve usted el control de la terminal y se reanude la ejecución de la tarea. O bien, una tarea detenida se puede ejecutar en el primer plano, lo cual le regresa el control de la terminal a ese programa.

Para colocar una tarea en ejecución en el segundo plano, teclee

```
[yyang@serverA ~]$ bg number
```

en donde **number** es el número de la tarea que quiere pasar al segundo plano.

NOTA Puede pasar a segundo plano cualquier proceso, si lo desea. Las aplicaciones que requieren entrada o salida por terminal se pondrán en un estado de detenidas, si las pasa a segundo plano. Por ejemplo, puede tratar de ejecutar la utilidad **top** en el segundo plano, al teclear **top &**. A continuación, verifique el estado de esa tarea con el comando **jobs**.

Variables de entorno

Cada caso de un shell, y cada programa que se esté ejecutando, tiene su propio “entorno”: ajustes que le dan un aspecto, sensación y, en algunos casos, comportamiento particulares. Por lo general, estos ajustes se controlan por medio de variables de entorno. Algunas variables de entorno tienen significados especiales para el shell, pero nada hay que impida a usted definir las propias y usarlas para sus propias necesidades. Es a través del uso de las variables de entorno que la mayor parte de los scripts de shells pueden realizar cosas interesantes y recordar los resultados de las entradas de los usuarios así como de las salidas de programas. Si ya está familiarizado con el concepto de variables de entorno en Windows NT/200x/XP, encontrará que muchas de las cosas que sabe acerca de ellas también se aplicarán a Linux, la única diferencia es la manera en que se fijan, se visualizan y se eliminan.

Impresión de variables de entorno

Para obtener una lista de sus variables de entorno, use el comando **printenv**. Por ejemplo,

```
[yyang@serverA ~]$ printenv  
HOSTNAME=serverA.example.org  
SHELL=/bin/bash  
TERM=xterm  
HISTSIZE=1000  
...<OUTPUT TRUNCATED>...
```

Para mostrar una variable específica de entorno, especifique la variable como un parámetro para **printenv**. Por ejemplo, enseguida se da el comando para ver la variable de entorno TERM:

```
[yyang@serverA ~]$ printenv TERM  
xterm
```

Fijación del valor de las variables de entorno

Para fijar el valor de una variable de entorno, use el formato siguiente:

```
[yyang@serverA ~]$ variable=value
```

en donde **variable** es el nombre de la variable y **value** es el valor que quiere asignarle. Por ejemplo, para fijar la variable de entorno FOO en el valor BAR, teclee

```
[yyang@serverA ~]$ FOO=BAR
```

Siempre que fije los valores de las variables de entorno de esta manera, permanecen locales para el shell en ejecución. Si quiere que el valor se pase a otros procesos que lance, use el comando integrado **export**. El formato del comando **export** es como sigue:

```
[yyang@serverA ~]$ export variable
```

en donde **variable** es el nombre de la variable. En el ejemplo de fijación del valor de la variable FOO, haría entrar el comando:

```
[yyang@serverA ~]$ export FOO
```



SUGERENCIA Puede combinar los pasos para fijar el valor de una variable de entorno con el comando **export**, de este modo: [yyang@serverA ~]\$ **export FOO=BAR**.

Si el valor de la variable de entorno que quiere fijar tiene espacios en él, ponga la variable entre comillas. Usando el ejemplo antes dado, para fijar FOO en "Welcome to the BAR of FOO.", haría entrar

```
[yyang@serverA ~]$ export FOO="Welcome to the BAR of FOO."
```

Entonces puede usar el comando **printenv** para ver el valor de la variable FOO que acaba de fijar, al teclear

```
[yyang@serverA ~]$ printenv FOO  
Welcome to the BAR of FOO.
```

Eliminación de las variables de entorno

Para eliminar una variable de entorno, use el comando **unset**. La sintaxis del comando **unset** es

```
[yyang@serverA ~]$ unset variable
```

en donde **variable** es el nombre de la variable que desea eliminar. Por ejemplo, el comando para eliminar la variable de entorno FOO es

```
[yyang@serverA ~]$ unset FOO
```

NOTA En esta sección, se supone que usted está usando BASH. Existen muchos otros shells de los cuales elegir; las alternativas más populares son el shell C (csh) y su hermano el shell Tenex/Turbo/Trusted C (tcsh), en los cuales se usan diferentes mecanismos para tener y fijar los valores de las variables de entorno. En este texto, documentamos BASH porque es el shell predeterminado de todas las nuevas cuentas Linux en la mayor parte de las distribuciones de éste.

Tuberías

Las *tuberías* son un mecanismo mediante el cual la salida de un programa se puede enviar como la entrada hacia otro programa. Programas separados se pueden encadenar entre sí para convertirse en herramientas extremadamente poderosas.

Usemos el programa **grep** para dar un ejemplo sencillo de cómo se pueden utilizar las tuberías. Dada una corriente de entrada, la utilidad **grep** tratará de correlacionar la línea con el parámetro que se le suministre y presentar sólo líneas correlativas. El lector recordará, de lo visto en la sección anterior, que el comando **printenv** imprime todas las variables de entorno. La lista que imprime puede ser larga, de modo que si, por ejemplo, deseara buscar todas las variables de entorno que contengan la cadena "TERM", podría teclear este comando:

```
[yyang@serverA ~]$ printenv | grep TERM
TERM=xterm
```

El carácter de barra vertical (|) representa la tubería entre **printenv** y **grep**.

Bajo Windows, el comando shell también utiliza la función de tubería. La diferencia primaria es que todos los comandos en una tubería de Linux se ejecutan en forma concurrente, en tanto que en Windows cada programa se ejecuta en orden, usando archivos temporales para contener los resultados intermedios.

Redirección

A través de la *redirección*, puede tomar la salida de un programa y hacer que se envíe en forma automática a un archivo (¡recuerde que en UNIX todo se considera como archivo!) En lugar del programa, el propio shell maneja este proceso, proporcionando de este modo un mecanismo estándar para la realización de la tarea (¡usar redirección es mucho más fácil que tener que recordar cómo hacer esto para cada uno de los programas!).

La redirección viene en tres clases: salida hacia un archivo, anexar a un archivo y enviar un archivo como entrada.

Para recoger la salida de un programa en un archivo, finalice la línea de comandos con el símbolo de mayor que (>) y el nombre del archivo hacia el cual desea redirigir la salida. Si está redirigiendo hacia un archivo existente y desea anexar datos adicionales a él, use dos símbolos > (>>) seguidos por el nombre del archivo. Por ejemplo, a continuación se da el comando para recoger la salida de la producción de una lista de directorios en un archivo llamado **/tmp/directory_listing**:

```
[yyang@serverA ~]$ ls > /tmp/directory_listing
```

Al continuar este ejemplo con la lista de directorios, podría anexar la cadena "Directory Listing" al final del archivo /tmp/directory_listing al teclear este comando:

```
[yyang@serverA ~] $ echo "Directory Listing" >> /tmp/directory_listing
```

La tercera clase de redirección, usando un archivo como entrada, se hace con el signo de menor que (<) seguido por el nombre del archivo. Por ejemplo, enseguida se da el comando para alimentar el archivo /etc/passwd en el programa grep:

```
[yyang@serverA ~] $ grep 'root' < /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin
```

ATAJOS DE LA LÍNEA DE COMANDOS

La mayor parte de los shells populares de UNIX/Linux tiene un número tremendo de atajos. Aprender los atajos y usarlos puede ser un enorme choque cultural para los usuarios que vienen del mundo de Windows. En esta sección, se explican los más comunes de los atajos de BASH y sus comportamientos.

Expansión de los nombres de archivo

Bajo los shells basados en UNIX/Linux, como BASH, los comodines en la línea de comandos se expanden antes de pasarse como un parámetro a la aplicación. Esto se encuentra en brusco contraste con el modo predeterminado de operación para las herramientas basadas en DOS, las cuales a menudo tienen que realizar su propia expansión de los comodines. El método de UNIX también significa que debe tener cuidado en dónde usa los caracteres comodines.

Los propios caracteres comodines en BASH son idénticos a los que se encuentran en command.com: el asterisco (*) se correlaciona con todos los nombres de archivo y el símbolo de cierre de interrogación (?) se correlaciona con los caracteres sencillos. Si necesita usar estos caracteres como parte de otro parámetro, por cualquier razón, se le puede escapar hacerlos preceder con un carácter de diagonal izquierda (\). Esto hace que el shell interprete el asterisco y el signo de interrogación como caracteres comunes y no como comodines.

NOTA La mayor parte de la documentación de UNIX se refiere a los comodines como expresiones comunes. La distinción es importante, ya que las expresiones comunes son sustancialmente más poderosas que justo los comodines solos. Todos los shells que vienen con Linux soportan expresiones comunes. Puede leer más acerca de ellos en la página de manual del shell (por ejemplo, man bash, man csh, man tcsh).

Variables de entorno como parámetros

Bajo BASH, puede usar variables de entorno como parámetros en la línea de comandos (aunque el mensaje de comandos de Windows también puede hacer esto, no es práctica común y, por consiguiente, es una convención que con frecuencia se olvida). Por ejemplo, la emisión del parámetro \$FOO hará que el valor de la variable de entorno FOO se pase en lugar de la cadena "\$FOO".

Comandos múltiples

Bajo BASH, se pueden ejecutar comandos múltiples en la misma línea, separándolos con puntos y comas (;). Por ejemplo, para ejecutar esta secuencia separada de comandos (**cat** y **ls**) en una sola línea:

```
[yyang@serverA ~]$ ls -l  
[yyang@serverA ~]$ cat /etc/passwd
```

en lugar de lo anterior, pudo teclear lo siguiente:

```
[yyang@serverA ~]$ ls -l ; cat /etc/passwd
```

Ya que el shell también es un lenguaje de programación, puede ejecutar comandos en serie sólo si el primero de ellos tiene éxito. Por ejemplo, use el comando **ls** para intentar hacer una lista de archivos que *no* exista en su directorio inicial y, enseguida, ejecute el comando **date** precisamente después del anterior, en la misma línea. Teclee

```
[yyang@serverA ~]$ ls does-not-exist.txt && date  
ls: does-not-exist.txt: No such file or directory
```

Este comando ejecutará el **ls**, pero ese comando fallará porque el archivo del que está tratando de hacer la lista no existe y, por lo tanto, de este modo el comando **date** tampoco se ejecutará. Pero si cambia el orden de los comandos, advertirá que el comando **date** tendrá éxito en tanto que el **ls** fallará. Intente

```
[yyang@serverA ~]$ date && ls does-not-exist.txt  
Mon Mar  7 22:41:13 PST 2030  
ls: does-not-exist.txt: No such file or directory
```

Acentos inversos

¿Qué le parece esta locura?: Puede tomar la salida de un programa y hacer que sea el parámetro de otro programa. ¿Suena extraño? Bien, tómese su tiempo para acostumbrarse a él: ésta es una de las características más útiles e innovadoras de la que se dispone en todos los shells de UNIX.

Los acentos inversos (`) le permiten embotar comandos como parámetros para otros comandos. En este libro, y en varios scripts del sistema, verá usada con frecuencia esta técnica.

Por ejemplo, se puede pasar el valor de un número (el número de ID de un proceso) a que se almacene en un archivo y pasar ese número al comando **kill**. Un ejemplo típico de esto se tiene cuando se necesita anular el servidor DNS, **named**. Cuando se inicia **named**, escribe su número de identificación del proceso (PID) en el archivo **/var/run/named/named.pid**. Por tanto, la manera genérica de anular el proceso **named** es mirar el número almacenado en **/var/run/named/named.pid**, usando el comando **cat**, y enseguida emitir el comando **kill** con ese valor. Por ejemplo,

```
[yyang@serverA ~]$ cat /var/run/named/named.pid  
253  
[yyang@serverA ~]$ kill 253
```

Un problema con la anulación del proceso **named** de esta manera es que no se puede automatizar; estamos contando con el hecho de que una persona leerá el valor en **/var/run/named/named.pid** para anular el número. Otro aspecto no es tanto un problema como una molestia: se necesitan dos pasos para detener el servidor DNS.

No obstante, usando acentos inversos podemos combinar los pasos en uno y hacerlo de una manera que se pueda automatizar. La versión con acentos inversos se observaría como esto:

```
[yyang@serverA ~]$ kill `cat /var/run/named/named.pid`
```

Cuando BASH ve este comando, primero ejecutará **cat /var/run/named/named.pid** y almacenará el resultado. A continuación, ejecutará **kill** y pasará el resultado almacenado a él. Desde nuestro punto de vista, esto sucede en un paso elegante.

NOTA Hasta ahora, en este capítulo, hemos analizado las características que son internas a BASH (o características integradas de BASH, como a veces se les llama). En el resto de este capítulo, se examinan varios comandos comunes accesibles por fuera de BASH.

HERRAMIENTAS PARA LA DOCUMENTACIÓN

Linux viene con dos herramientas excelentemente útiles para hacer que la documentación sea accesible: **man** e **info**. En la actualidad, existe una gran cantidad de traslapos entre estos dos sistemas de documentación porque en muchas aplicaciones se está moviendo su documentación hacia el formato **info**. Este formato se considera superior a **man** porque permite establecer hipervínculos entre la documentación de una manera semejante a la de la Web, pero sin tener que escribirse en realidad en el formato HTML.

Por otra parte, el formato **man** ha estado por allí durante décadas. Para millares de utilidades, sus páginas man (abreviatura de *manual*) constituyen su única documentación. Además, en muchas aplicaciones se continúa utilizando el formato **man** debido a que en muchos otros sistemas operativos semejantes a UNIX (como Sun Solaris) se usa ese formato.

Los dos sistemas de documentación, **man** e **info**, estarán por allí durante un largo tiempo. Se recomienda con intensidad que se sienta cómodo con los dos.

SUGERENCIA En muchas distribuciones de Linux también se incluye una gran cantidad de documentación en el directorio **/usr/doc** o en el **/usr/share/doc**.

El comando **man**

Se mencionó con bastante anterioridad en este libro que las páginas man son documentos que se encuentran en línea (o en el sistema local) y que cubren el uso de herramientas y su configuración correspondiente. El formato del comando **man** es como sigue:

```
[yyang@serverA ~]$ man program_name
```

en donde **program_name** identifica el programa en el que está usted interesado. Por ejemplo, para ver la página man para la utilidad **ls** que hemos estado usando, teclee

```
[yyang@serverA ~]$ man ls
```

Mientras está leyendo acerca de fuentes de información de UNIX y relacionadas con UNIX (grupos de noticias, etc.), puede ser que encuentre referencias a comandos seguidos por números entre paréntesis; por ejemplo, **ls(1)**. El número representa la sección de las páginas del manual (vea la tabla 5-1). En cada sección, se cubren varias áreas de la materia, para dar cabida al hecho de que algunas herramientas (como **printf**) son comandos/funciones en el lenguaje de programación C así como comandos en las líneas de comandos.

Para referirse a una sección man específica, sencillamente especifique el número de sección como el primer parámetro y enseguida el comando como el segundo parámetro. Por ejemplo, para obtener la información de los programadores C sobre **printf**, introducirá lo siguiente:

```
[yyang@serverA ~]$ man 3 printf
```

Para obtener la información de la línea de comandos, teclearía esto:

```
[yyang@serverA ~]$ man 1 printf
```

Si no especifica un número de sección con el comando **man**, el comportamiento predeterminado es que se imprima primero el menor de los números de sección.

Desgraciadamente, a veces esta organización puede ser difícil de usar y, como resultado, existen varias otras alternativas disponibles.

Sección del manual	Tema
1	Herramientas del usuario
2	Llamadas del sistema
3	Llamadas de biblioteca C
4	Información de controladores de dispositivos
5	Archivos de configuración
6	Juegos
7	Paquetes
8	Herramientas del sistema

Tabla 5-1. Secciones de la página man

SUGERENCIA Una opción práctica para el comando **man** es **-f** precediendo al parámetro del mismo. Con esta opción, **man** buscará la información resumen de todas las páginas man y hará una lista de las páginas que se correlacionen con el comando que haya usted especificado, junto con su número de sección. Por ejemplo,

```
[yyang@serverA ~] $ man -f printf
asprintf          (3) - print to allocated string
printf            (1) - format and print data
printf            (3) - formatted output conversion
```

El sistema texinfo

Otra forma común de la documentación es texinfo. Establecida como el estándar de GNU, texinfo es un sistema de documentación semejante al formato de hipervínculos de la World Wide Web. Debido a que los documentos se pueden enlazar entre sí por hipervínculos, texinfo con frecuencia es más fácil de leer, usar y buscar que las páginas man.

Para leer los documentos texinfo en una herramienta o aplicación específicas, llame **info** con el parámetro que especifique el nombre de la herramienta. Por ejemplo, para leer acerca del programa **grub**, teclee

```
[yyang@serverA ~] $ info grub
```

En general, querrá verificar si existe una página man, antes de usar **info** (todavía hay una mayor cantidad de información en el formato **man** que en el texinfo). Por otra parte, en algunas páginas man se dirá en forma explícita que las páginas texinfo contienen información más autorizada y que deben leerse en su lugar.

COMPRENSIÓN DE LA FORMACIÓN DE LISTAS DE ARCHIVOS, DE LAS PROPIEDADES Y DE LOS PERMISOS

La administración de archivos bajo Linux es diferente a la misma acción bajo Windows NT / 200x / XP y radicalmente distinta de la que se realiza bajo Windows 95 / 98. En esta sección, discutiremos las herramientas básicas de la administración de archivos, para Linux. Empezaremos con aspectos específicos sobre algunos comandos útiles para fines generales y, después, retrocederemos y analizaremos alguna información básica.

Formar listas de archivos: ls

El comando **ls** se usa para hacer una lista de todos los archivos que se encuentran en un directorio. De más de 26 opciones de las que se dispone, aquellas cuya lista se da en la tabla 5-2 son las de uso más común. Las opciones se pueden usar en cualquier combinación.

Opción para ls	Descripción
-l	Formación de listas largas. Además del nombre del archivo, se muestra el tamaño de éste, fecha/hora, permisos, propiedad e información del grupo.
-a	Todos los archivos. Muestra todos los archivos que se encuentran en el directorio, incluyendo archivos escondidos. Los nombres de estos últimos archivos empiezan con un punto.
-t	Hace una lista en orden del momento de la última modificación.
-r	Invierte la lista.
-1	Lista en una sola columna.
-R	Hace una lista en forma recursiva de todos los archivos y subdirectorios.

Tabla 5-2. Opciones comunes de ls

Para hacer una lista de todos los archivos que se encuentran en un directorio obteniendo una lista larga, teclee este comando:

```
[yyang@serverA ~] $ ls -la
```

Para hacer una lista de todos los archivos no escondidos de un directorio que empiecen con la letra A, teclee esto:

```
[yyang@serverA ~] $ ls A*
```

SUGERENCIA Linux/UNIX es muy sensible a las mayúsculas y minúsculas. Por ejemplo, un archivo nombrado “**thefile.txt**” es muy diferente de uno nombrado “**Thefile.txt**”.

Si no existe ese archivo en su directorio de trabajo, **ls** imprime un mensaje que se lo dice.

Tipos de archivos y de directorios

Bajo Linux (y UNIX en general), casi todo se resume a un archivo. Originalmente, esto se hizo para simplificar el trabajo del programador. En lugar de tener que comunicarse en forma directa con los controladores de dispositivos, se usan archivos especiales (los cuales aparecen como archivos ordinarios para la aplicación) como un puente. Varios tipos de archivos dan lugar a todos estos usos de los mismos.

Archivos normales

Los archivos normales son sólo eso: normales. Contienen datos o ejecutables, y el sistema operativo no hace suposiciones acerca de su contenido.

Directorio

Los archivos directorios son un caso especial de archivos normales. En los archivos directorios se hace una lista de las ubicaciones de otros archivos, algunos de los cuales pueden ser otros directorios (esto es semejante a las carpetas de Windows). En general, el contenido de los archivos directorios no tendrá importancia para las operaciones diarias de usted, a menos que necesite abrir y leer el archivo por sí mismo, en lugar de usar las aplicaciones existentes para navegar en los directorios (esto sería semejante a tratar de leer en forma directa la tabla de asignación de archivos de DOS, en lugar de usar command.com, para navegar en los directorios, o las llamadas del sistema findfirst/findnext).

Vínculos fuertes

En el sistema de archivos de Linux, cada archivo tiene su propio nodo i. Este tipo de nodos se mantienen al día de los atributos de un archivo y de su ubicación en el disco. Si necesita ser capaz de referirse a un solo archivo usando dos nombres separados de archivo, puede crear un vínculo fuerte. El *vínculo fuerte* tendrá el mismo nodo i que el archivo original y, por lo tanto, se observará y se comportará precisamente como el original. Con cada vínculo fuerte que se cree, se incrementa una cuenta de referencia. Cuando se elimina uno de esos vínculos, se disminuye esa cuenta. Hasta que la cuenta de referencia llegue a cero, el archivo permanecerá en el disco.

NOTA No puede existir un vínculo fuerte entre dos archivos en particiones separadas. Esto se debe a que este tipo de vínculo se refiere al archivo original por el nodo i, y el nodo i de un archivo puede diferir entre los sistemas de archivos.

Vínculos simbólicos

A diferencia de los vínculos fuertes, los cuales apuntan a un archivo por su nodo i, un *vínculo simbólico* apunta a otro archivo por su nombre. Esto permite a este último tipo de vínculo (a menudo abreviado como symlinks) apuntar a archivos ubicados en otras particiones, incluso en otras unidades de la red.

Dispositivos de bloques

Ya que se tiene acceso a todos los controladores de dispositivos a través del sistema de archivos, se usan los archivos del tipo *dispositivo de bloques* para establecer la interfaz con dispositivos como los discos. Un dispositivo de bloques tiene tres rasgos identificadores:

- ▼ Tiene un número mayor.
- Tiene un número menor.
- ▲ Cuando se ve usando el comando `ls -l`, muestra b como el primer carácter del campo de permisos.

Por ejemplo,

```
[yyang@serverA ~]$ ls -l /dev/hda
brw-rw---- 1 root disk 3, 0 Mar 7 13:35 /dev/hda
```

Note la b al principio de los permisos del archivo; el 3 es el número mayor y 0 es el menor.

El número mayor de un archivo de dispositivo de bloques identifica el controlador representado de dispositivo. Cuando se tiene acceso a este archivo, el número menor se pasa al controlador como parámetro, diciéndole a cuál dispositivo se está teniendo acceso. Por ejemplo, si se tienen dos puertos en serie, compartirán el mismo controlador de dispositivo y, por consiguiente el mismo número mayor, pero cada puerto en serie tendrá un número menor único.

Dispositivos de caracteres

Semejantes a los dispositivos de bloques, los *dispositivos de caracteres* son archivos especiales que le permiten tener acceso a los dispositivos a través del sistema de archivos. La diferencia obvia entre los dispositivos de bloques y los de caracteres es que los de bloques se comunican con los dispositivos reales en bloques grandes, en tanto que los de caracteres trabajan un carácter a la vez (un disco duro es un dispositivo de bloques; un módem es un dispositivo de caracteres). Los permisos de los dispositivos de caracteres empiezan con una *c*, y el archivo tiene un número mayor y uno menor. Por ejemplo,

```
[yyang@serverA ~]$ ls -l /dev/ttys0
crw-rw---- 1 root uucp 4, 64 Mar 7 21:36 /dev/ttys0
```

Tuberías nombradas

Las *tuberías nombradas* son un tipo especial de archivo que da lugar a la comunicación entre procesos. Si se usa el comando **mknod** (que se discute más adelante en este capítulo), puede crear un archivo de tubería nombrada que un proceso puede abrir para leer y otro proceso puede abrir para escribir, permitiendo de esta manera que los dos se comuniquen entre sí. Esto funciona especialmente bien cuando un programa rechaza tomar la entrada de una tubería de línea de comandos, pero otro programa necesita alimentar al otro un dato y usted no cuenta con el espacio de disco para un archivo temporal.

Para un archivo de tubería nombrada, el primer carácter de sus permisos de archivo es una *p*. Por ejemplo, si existe una tubería nombrada que se llama mypipe en su directorio actual de trabajo (*pwd*), una lista larga del archivo de tubería nombrada mostraría esto:

```
[yyang@serverA ~]$ ls -l mypipe
prw-r--r-- 1 root      root          0 Mar 16 10:47 mypipe
```

Cambiar la propiedad: chown

El comando **chown** le permite cambiar la propiedad de un archivo a alguien más. Sólo el usuario raíz puede hacer esto (es posible que los usuarios normales no quiten o roben la propiedad de otro usuario). El formato del comando es como sigue:

```
[root@serverA ~]# chown [-R] username filename
```

en donde **username** es el nombre para entrar del usuario a quien desea asignar la propiedad y **filename** es el nombre del archivo en cuestión. El nombre de archivo también puede ser un directorio.

Se aplica la opción **-R** cuando el nombre especificado de archivo es un nombre de directorio. Esta opción le dice al comando que descienda en forma recursiva a través del árbol de directorios y que aplique la nueva propiedad no sólo al propio directorio, sino a todos los archivos y directorios que se encuentren dentro de él.

NOTA Linux le permite usar una notación especial con **chown** para también suministrar los archivos para el grupo a **chgrp**. El formato del comando se convierte en **chown username.groupname filename**.

Cambiar el grupo: chgrp

La utilidad **chgrp** de la línea de comandos le deja cambiar los ajustes de grupo de un archivo. Funciona de modo muy semejante a **chown**. Enseguida se da el formato:

```
[root@serverA ~]# chgrp [-R] groupname filename
```

en donde **groupname** es el nombre del grupo al cual desea asignar la propiedad del nombre de archivo. El nombre de archivo también puede ser un directorio.

Se aplica la opción **-R** cuando el nombre especificado de archivo es un nombre de directorio. Como con **chown**, esta opción le dice al comando que descienda en forma recursiva a través del árbol de directorios y que aplique la nueva propiedad no sólo al propio directorio, sino también a todos los archivos y directorios que se encuentren dentro de él.

Cambiar el modo: chmod

Dentro del sistema de Linux, los directorios y archivos tienen permisos asociados con ellos. De manera predeterminada, los permisos los fija el propietario del archivo, el grupo asociado con este último y todos los demás que tienen acceso al mismo (también conocidos como propietario, grupo, otros). Cuando haga una lista de archivos o directorios, vea los permisos en la primera columna de la salida. Los permisos se dividen en cuatro partes. La primera se representa por medio del primer carácter del permiso. Los archivos normales no tienen valor especial y se representan con un carácter de guión (-). Si el archivo tiene un atributo especial, se representa por medio de una letra. Los dos atributos especiales que más nos interesan aquí son los directorios (**d**) y los vínculos simbólicos (**1**).

La segunda, tercera y cuarta partes de un permiso se representan en tres trozos de caracteres. La primera parte indica el permiso del propietario del archivo. La segunda indica el permiso del grupo. La última indica el permiso del mundo. En el contexto de UNIX, por “mundo” se entiende todos los usuarios del sistema, sin importar los valores fijados de su grupo.

Las que siguen son las letras que se usan para representar los permisos y sus valores correspondientes. Cuando combine los atributos, agregue sus valores. Se usa el comando **chmod** para fijar los valores de los permisos.

Letra	Permiso	Valor
R	Leer	4
W	Escribir	2
X	Ejecutar	1

Si se usa el modo de comandos numéricos, por lo general se conocen como los permisos *octales*, ya que el valor puede variar desde 0 hasta 7. Para cambiar los permisos en un archivo, simplemente agregue estos valores juntos para cada permiso que deseé aplicar.

Por ejemplo, si quiere hacerlo de modo que sólo el usuario (propietario) pueda tener pleno acceso (RWX) al archivo llamado **foo**, teclearía

```
[yyang@serverA ~]$ chmod 700 foo
```

Lo que es importante hacer notar es que usando el modo octal siempre *reemplaza* cualesquier permisos que estuvieran fijados. De modo que si hubo un archivo en **/usr/local** que fue SetUID y ejecutó el comando **chmod -R 700 /usr/local**, ese archivo ya no será SetUID. Si quiere cambiar ciertos bits, debe usar el modo simbólico de **chmod**. Este modo resulta ser más fácil de recordar y puede agregar, restar o sobreescribir permisos.

La forma simbólica de **chmod** le permite fijar los bits del propietario, del grupo o de los otros. También puede fijar los bits para todos. Por ejemplo, si quiere cambiar un archivo llamado **foobar.sh** de modo que sea ejecutable por el propietario, puede ejecutar el comando siguiente:

```
[yyang@serverA ~]$ chmod u+x foobar.sh
```

Si también quiere cambiar el bit del grupo para ejecutar, use lo siguiente:

```
[yyang@serverA ~]$ chmod ug+x foobar.sh
```

Si necesita especificar permisos diferentes para los otros, sólo agregue una coma y sus símbolos de permisos como se dan enseguida:

```
[yyang@serverA ~]$ chmod ug+x,o-rwx foobar.sh
```

Si no desea agregar o restar un bit de permiso, puede usar el signo **=**, en lugar de **+** o **-**. Con esto se escribirán los bits específicos para el archivo y se borrará cualquier otro bit para ese permiso. En los ejemplos anteriores, usamos **+** para agregar el bit de ejecutar a los campos de usuario y del grupo. Si sólo quiere el bit de ejecutar, reemplazaría el **+** con **=**. También existe un cuarto carácter que puede usar: **a**. Con éste se aplicarán los bits de permisos a todos los campos.

En la lista que sigue, se muestran las combinaciones más comunes de los tres permisos. Existen otras combinaciones, como **-wx**. Pero rara vez se usan.

Letra	Permiso	Valor
---	No permisos	0
r--	Sólo leer	4
rw-	Leer y escribir	6
rwx	Leer, escribir y ejecutar	7
r-x	Leer y ejecutar	5
--x	Sólo ejecutar	1

Para archivo, se agrupan juntos tres de estos trozos de tres letras. El primer trozo representa los permisos para el propietario del archivo; el segundo los permisos para el grupo de este último, y el último representa los permisos para todos los usuarios en el sistema. En la tabla 5-3 se muestran algunas combinaciones de permisos, sus equivalentes numéricos y sus descripciones.

Permiso	Equivalente numérico	Descripción
-rw-----	600	El propietario tiene permisos de leer y escribir.
-rw-r--r--	644	El propietario tiene permisos de leer y escribir; el grupo y el mundo tienen permiso sólo de leer.
-rw-rw-rw-	666	Todos tienen permisos de leer y escribir. No recomendado; esta combinación permite que cualquiera pueda tener acceso al archivo y cambiarlo.
-rwx-----	700	El propietario tiene permisos de leer, escribir y ejecutar. La mejor combinación para programas o ejecutables que el propietario deseé activar.
-rwxr-xr-x	755	El propietario tiene permisos de leer, escribir y ejecutar. Todos los demás tienen permisos de leer y ejecutar.
-rwxrwxrwx	777	Todos tienen los privilegios de leer, escribir y ejecutar. Como el ajuste 666, esta combinación debe evitarse.
-rwx---x--x	711	El propietario tiene permisos de leer, escribir y ejecutar; todos los demás tienen permiso sólo de ejecutar. Útil para programas que usted desea que los otros ejecuten pero no copien.
drwx-----	700	Éste es un directorio creado con el comando mkdir . Sólo el propietario puede leer y escribir en este directorio. Note que todos los directorios deben tener el valor del bit de ejecutable.
drwxr-xr-x	755	Este directorio sólo puede ser cambiado por el propietario, pero todos los demás pueden ver su contenido.
drwx---x--x	711	Una combinación práctica para conservar un directorio legible para el mundo, pero restringido respecto al acceso por el comando ls . Un archivo sólo puede ser leído por alguien que conoce el nombre del mismo.

Tabla 5-3. Permisos para los archivos

ADMINISTRACIÓN Y MANIPULACIÓN DE ARCHIVOS

En esta sección, se cubren las herramientas básicas de la línea de comandos para administrar archivos y directorios. La mayor parte de esto será familiar para cualquiera que haya usado una interfaz de líneas de comandos: las mismas antiguas funciones, pero nuevos comandos para ejecutar.

Copiar archivos: cp

El comando **cp** se usa para copiar archivos. Tiene un número sustancial de opciones. Vea su página man para obtener detalles adicionales. De manera predeterminada, este comando funciona en forma silenciosa, sólo presenta información del estado si se presenta una condición de error. Las que siguen son las opciones más comunes para **cp**:

Opción para cp	Descripción
-f	Fuerza la copia; no pide verificación
-I	Copia interactiva; antes de que se copie cada archivo, verifica con el usuario

En primer lugar, usemos el comando **touch** para crear un archivo vacío, llamado **foo.txt**, en el directorio inicial del usuario yyang. Teclee

```
[yyang@serverA ~]$ touch foo.txt
```

Para usar el comando **cp** (copiar) para copiar **foo.txt** en el **foo.txt.html**. Teclee

```
[yyang@serverA ~]$ cp foo.txt foo.txt.html
```

Para copiar en forma interactiva todos los archivos que finalizan **.html** en el directorio **/tmp**, teclee este comando:

```
[yyang@serverA ~]$ cp -i *.html /tmp
```

Mover archivos: mv

Se usa el comando **mv** para mover archivos de una ubicación a otra. Los archivos también se pueden mover a través de sistemas de particiones/archivos. Mover archivos a través de particiones comprende una operación de copia y, como resultado, el comando mover puede tardar más tiempo. Pero el lector encontrará que mover archivos dentro del mismo sistema de archivos es casi instantáneo. Las que siguen son las opciones más comunes para **mv**:

Opción para mv	Descripción
-f	Fuerza el movimiento
-I	Movimiento interactivo

Para mover un archivo nombrado **foo.txt.html** de **/tmp** hasta el directorio actual de trabajo de usted, use este comando:

```
[yyang@serverA ~] $ mv /tmp/foo.txt.html .
```

NOTA El punto precedente (.) no es un error tipográfico; literalmente significa “este directorio”.

No existe una herramienta explícita para renombrar, de modo que puede usar el comando **mv**. Para renombrar el archivo **foo.txt.html** como **foo.txt.htm**, teclee

```
[yyang@serverA ~] $ mv foo.txt.html foo.txt.htm
```

Vincular archivos: ln

El comando **ln** le permite establecer vínculos fuertes y vínculos suaves (vea “Tipos de archivos y de directorios” que está antes en este capítulo). El formato general de **ln** es como sigue:

```
[yyang@serverA ~] $ ln original_file new_file
```

Aunque **ln** tiene muchas opciones, rara vez las necesitará para usar la mayor parte de ellas. Con la opción más común, **-s**, se crea un vínculo simbólico, en lugar de uno fuerte.

Para crear un vínculo simbólico llamado **link-to-foo.txt** que apunta al archivo original llamado **foo.txt**, emita el comando

```
[yyang@serverA ~] $ ln -s foo.txt link-to-foo.txt
```

Encontrar un archivo: find

El comando **find** le permite buscar archivos según varios criterios. Como las herramientas que ya hemos discutido, **find** tiene un gran número de opciones acerca de las cuales puede leer en su página man. Enseguida está el formato general de **find**:

```
[yyang@serverA ~] $ find start_dir [options]
```

en donde **start_dir** es el directorio a partir del cual se debe iniciar la búsqueda.

Para encontrar todos los archivos en el directorio actual de usted (es decir, en el directorio **“.”**) al que no ha entrado en al menos siete días, use el comando que sigue:

```
[yyang@serverA ~] $ find . -atime 7
```

Teclee este comando para hallar todos los archivos que se encuentran en su directorio actual de trabajo, cuyos nombres son **core** y, a continuación, bórrelos (es decir, ejecute en forma automática el comando **rm**):

```
[yyang@serverA ~] $ find . -name core -exec rm {} \;
```

SUGERENCIA La sintaxis para la opción **-exec** con el comando **find**, como se usa aquí, a veces puede requerir un poco tratar de recordarlo y, por consiguiente, también puede usar el método **xargs**, en lugar de la opción **exec** usada en este ejemplo. Si se usa **xargs**, el comando entonces se escribiría

```
[yyang@serverA ~] $ find . -name 'core' | xargs rm
```

Para hallar todos los archivos en su pwd, cuyos nombres finalizan en **.txt** (es decir, archivos que tienen la extensión **txt**) y que también tienen un tamaño menor de 100K, emita este comando:

```
[yyang@serverA ~]$ find . -name '*.txt' -size -100k
```

Para hallar todos los archivos en su pwd, cuyos nombres finalizan en **.txt** (es decir, archivos que tienen la extensión **txt**) y que también tienen un tamaño mayor de 100K, emita este comando:

```
[yyang@serverA ~]$ find . -name '*.txt' -size 100k
```

Compresión de archivos: gzip

En las distribuciones originales de UNIX, a la herramienta para comprimir archivos se le llamaba, de manera apropiada, **compress**. Por desgracia, el algoritmo fue patentado por alguien esperando ganar una gran cantidad de dinero. En lugar de pagar, la mayor parte de los sitios buscaron y hallaron otra herramienta de compresión con un algoritmo no patentado: **gzip**. Incluso mejor, **gzip** logra de manera uniforme mejores niveles de compresión que los que logra **compress**. Otro premio: cambios recientes han permitido que **gzip** descomprima archivos que fueron comprimidos con el uso del comando **compress**.

NOTA La extensión del nombre de archivo suele identificar un archivo comprimido con **gzip**. Por lo común, estos archivos finalizan en **.gz** (los archivos comprimidos con **compress** finalizan en **.z**).

Note que **gzip** comprime el archivo en su lugar, lo que significa que, después del proceso de compresión, se elimina el archivo original y lo único que queda es el archivo comprimido.

Para comprimir un archivo llamado **foo.txt.htm** en su pwd, teclee

```
[yyang@serverA ~]$ gzip foo.txt.htm
```

Y, a continuación, descomprimalo, use **gzip** de nuevo, con la opción **-d**:

```
[yyang@serverA ~]$ gzip -d foo.txt.htm.gz
```

Emita este comando para comprimir todos los archivos que finalizan en **.htm**, en su pwd, usando la mejor compresión posible:

```
[yyang@serverA ~]$ gzip -9 *.htm
```

bzip2

Si ha advertido que algunos archivos tienen una extensión **.bz**, éstos se han comprimido con la utilidad de compresión **bzip2**. En la herramienta **bzip2** se usa un algoritmo diferente de compresión que suele dar por resultado archivos más pequeños que los comprimidos con la utilidad **gzip**, pero se usa semántica que es semejante a la de esta última; para obtener más información,lea la página man en **bzip2**.

Crear un directorio: `mkdir`

El comando `mkdir` de Linux es idéntico al mismo comando en otras inclinaciones de UNIX, como también en MS-DOS. Una opción usada a menudo con el comando `mkdir` es la `-p`. Esta opción forzará a que `mkdir` cree directorios padres si todavía no existen. Por ejemplo, si necesita crear `/tmp/bigdir/subdir/mydir` y el único directorio que existe es el `/tmp`, el uso de `-p` hará que se creen en forma automática `bigdir` y `subdir`, junto con `mydir`.

Cree un árbol de directorios como `bigdir/subdir/finaldir` en su `pwd`. Teclee

```
[yyang@serverA ~]$ mkdir -p bigdir/subdir/finaldir
```

Para crear un solo directorio llamado `mydir`, use este comando:

```
[yyang@serverA ~]$ mkdir mydir
```

Eliminar un directorio: `rmdir`

El comando `rmdir` no ofrece sorpresas para quienes estén familiarizados con la versión de DOS de ese comando; sencillamente elimina un directorio existente. Este comando también acepta el parámetro `-p`, el cual también elimina los directorios padres.

Por ejemplo, si se quiere deshacer de todos los directorios, desde `bigdir` hasta `finaldir`, que se crearon con anterioridad, emitiría sólo este comando:

```
[yyang@serverA ~]$ rmdir -p bigdir/subdir/finaldir
```

Para eliminar un directorio llamado `mydir`, teclearía esto:

```
[yyang@serverA ~]$ rmdir mydir
```



SUGERENCIA También puede usar el comando `rm` con la opción `-r`, para borrar los directorios.

Mostrar el directorio actual de trabajo: `pwd`

Es inevitable que llegará a sentarse ante una estación de trabajo que ya se haya hecho entrar y usted no sabe en dónde se encuentra en el árbol de directorios. Para obtener esta información, necesita el comando `pwd`. Su única tarea es imprimir el directorio actual de trabajo.

Para presentar su directorio actual de trabajo, use este comando:

```
[yyang@serverA ~]$ pwd  
/home/yyang
```

Archivo de cinta: `tar`

Si está familiarizado con el programa PKZip, está acostumbrado al hecho de que la herramienta de compresión reduce el tamaño del archivo pero también consolida los archivos en archivos comprimidos. Bajo Linux, este proceso se separa en dos herramientas: `gzip` y `tar`.

El comando **tar** combina archivos múltiples en un solo archivo grande. Está separado de la herramienta de compresión, de modo que le permite seleccionar cuál herramienta de compresión quiere usar e, incluso, si desea esa compresión. De manera adicional, **tar** es capaz de leer los dispositivos y escribir en estos, lo que lo hace una buena herramienta para respaldar los dispositivos de cinta.

NOTA Aun cuando el nombre del programa **tar** incluye la palabra “cinta”, al crear archivos no es necesario leer en una unidad de cinta o escribir en ella. De hecho, rara vez usará **tar** con una unidad de cinta en las situaciones cotidianas (aparte del respaldo). La razón por la que se le nombró **tar** en principio fue que, cuando se creó originalmente, el espacio limitado en disco significaba que la cinta era el lugar más lógico para poner archivos. Por lo común, se usaría la opción **-f** en **tar** para especificar el archivo en dispositivo de cinta, en lugar de un archivo UNIX tradicional. Sin embargo, el lector debe estar consciente que todavía puede usar **tar** directo a un dispositivo.

La sintaxis para el comando **tar** es

```
[yyang@serverA ~] $ tar option... filename...
```

Enseguida se muestran algunas opciones para el comando **tar**:

Opción para tar	Descripción
-c	Crea un archivo nuevo
-t	Ve el contenido de un archivo
-x	Extrae el contenido de un archivo
-f	Especifica el nombre del archivo (o dispositivo) en el cual se ubica el archivo
-v	Proporciona descripciones amplias en el transcurso de las operaciones
-j	Filtrá el archivo a través de la utilidad de compresión bzip2
-z	Filtrá el archivo a través de la utilidad de compresión gzip

Para ver un uso muestra de la utilidad **tar**, en primer lugar cree una carpeta llamada **junk** en el **pwd** que contenga algunos archivos vacíos nombrados **1, 2, 3, 4**. Teclee

```
[yyang@serverA ~] $ mkdir junk ; touch junk/{1,2,3,4}
```

Ahora cree un archivo llamado **junk.tar** que contenga todos los archivos que se encuentran en la carpeta llamada **junk** que acaba de crear: teclee

```
[yyang@serverA ~] $ tar -cf junk.tar junk
```

Cree otro archivo llamado **2junk.tar** que contenga todos los archivos que se encuentran en la carpeta **junk**, pero en esta ocasión agregue la opción **-v** (descripción) para que se muestre lo que está sucediendo, a medida que sucede. Teclee lo siguiente:

```
[yyang@serverA ~]$ tar -vcf 2junk.tar junk
junk/
junk/4
junk/3
junk/1
junk/2
```

NOTA Debe notar que los archivos creados en estos ejemplos no están comprimidos de manera alguna. Los archivos y el directorio sólo se han combinado en un solo archivo.

Para crear un archivo comprimido con **gzip**, llamado **3junk.tar.gz**, que contenga todos los archivos de la carpeta **junk** y que se muestre lo que está sucediendo a medida que sucede, emita este comando:

```
[yyang@serverA ~]$ tar -cvzf 3junk.tar.gz junk
```

Para extraer el contenido del archivo **tar**, comprimido con **gzip**, creado aquí, así como la descripción acerca de lo que se está haciendo, emita el comando:

```
[yyang@serverA ~]$ tar -xvzf 3junk.tar.gz
```

SUGERENCIA El comando **tar** es una de las pocas utilidades de Linux en la que importa el orden con el cual especifica sus opciones. Si emitiera el comando **tar** que acaba de darse como
tar -xvfz 3junk.tar.gz
el comando fallará porque no se hace seguir inmediatamente a la opción **-F** por un nombre de archivo.

Si lo desea, también puede especificar un dispositivo físico para **tar**, hacia el cual ir y del cual salir. Esto resulta práctico cuando necesita transferir un conjunto de archivos de un sistema a otro y que, por alguna razón, no puede crear un sistema de archivos en el dispositivo (o, a veces, sólo es más entretenido hacerlo de esta manera). Para crear un archivo en el primer dispositivo de disquete (`/dev/fd0`), introduciría esto:

```
[yyang@serverA ~]$ tar -cvzf /dev/fd0 junk
```

NOTA El comando **tar -cvzf /dev/fd0** tratará al disco como un dispositivo virgen y borrará cualquier cosa que esté escrita en él.

Para extraer ese archivo de un disco, teclearía

```
[yyang@serverA ~]$ tar -xvzf /dev/fd0
```

Concatenar archivos: cat

El programa **cat** cumple con un papel en extremo sencillo: mostrar archivos. Se pueden hacer cosas más creativas con él, pero casi todo su uso estará en la forma de sencillamente presentar el contenido de archivos de texto; de modo muy semejante al comando **type** bajo DOS. Debido a que se pueden especificar múltiples nombres de archivos en la línea de comandos, es posible concatenar archivos en un solo archivo grande y continuo. Éste se diferencia de **tar** en que el archivo resultante no tiene información de control para mostrar las fronteras de los diferentes archivos.

Para presentar el archivo **/etc/passwd**, use este comando:

```
[yyang@serverA ~]$ cat /etc/passwd
```

Para desplegar el archivo **/etc/passwd** y el archivo **/etc/group**, emita este comando:

```
[yyang@serverA ~]$ cat /etc/passwd /etc/group
```

Teclee este comando para concatenar **/etc/passwd** con **/etc/group** y enviar la salida al archivo **users-and-groups.txt**:

```
[yyang@serverA ~]$ cat /etc/passwd /etc/group > users-and-groups.txt
```

Para anexar el contenido del archivo **/etc/hosts** al **users-and-groups.txt** que acaba de crear, teclee:

```
[yyang@serverA ~]$ cat /etc/hosts >> users-and-groups.txt
```

SUGERENCIA Si quiere para **cat** un archivo en orden inverso, puede usar el comando **tac**.

Presentar un archivo una pantalla a la vez: more

El comando **more** funciona en gran parte de la misma manera en que lo hace la versión del programa DOS. Toma un archivo de entrada y lo presenta una pantalla a la vez. El archivo de entrada puede venir de su **stdin** o de un parámetro de la línea de comandos. En la página man, se pueden hallar parámetros adicionales de la línea de comandos, aunque rara vez se usan.

Para ver el archivo **/etc/passwd** una pantalla a la vez, use este comando:

```
[yyang@serverA ~]$ more /etc/passwd
```

Para ver las listas de directorios generadas por el comando **ls** una pantalla a la vez, haga entrar:

```
[yyang@serverA ~]$ ls | more
```

Utilización del disco: du

Con frecuencia necesitará determinar en dónde se está consumiendo espacio de disco, y por quién, ¡en especial cuando usted lo está utilizando poco! El comando **du** le permite determinar la utilización del disco en términos de directorio por directorio.

Enseguida se dan algunas de las opciones de las que se dispone:

Opción para du	Descripción
-c	Produce un gran total al final de la ejecución.
-h	Imprime un formato legible para las personas.
-k	Imprime tamaños en kilobytes en lugar de en tamaño de bloques (nota: bajo Linux, un bloque es igual a 1K, pero esto no se cumple para todas las formas de UNIX).
-s	Resume. Sólo imprime un total para cada argumento.

Para presentar la cantidad total de espacio que se está usando por todos los archivos y directorios en su `pwd`, en formato legible para las personas, use este comando:

```
[yyang@serverA ~]$ du -sh .
2.2M
```

Mostrar la ubicación del directorio de un archivo: which

El comando `which` busca su trayectoria completa para hallar el nombre de un ejecutable especificado en la línea de comandos. Si se encuentra el archivo, la salida del comando incluye la trayectoria real del archivo.

Use el comando que sigue para averiguar en cuál directorio está ubicado el binario para el comando `rm`:

```
[yyang@serverA ~]$ which rm
/bin/rm
```

Puede ser que encuentre esto semejante al comando `find`. En este caso, la diferencia es que como `which` sólo busca la trayectoria es mucho más rápido. Por supuesto, también es mucho más limitado que `find`, pero si todo lo que usted está buscando es un programa, encontrará que es una mejor selección de comandos.

Localizar un comando: whereis

La herramienta `whereis` busca su trayectoria y presenta el nombre del programa y su directorio absoluto, el archivo fuente (si está disponible) y la página man para el programa (una vez más, si está disponible).

Para hallar la ubicación del programa, la fuente y la página del manual para el comando `grep`, use esto:

```
[yyang@serverA ~]$ whereis grep
grep: /bin/grep /usr/share/man/man1/grep.1.gz /usr/share/man/man1p/grep.1p.gz
```

Espacio libre del disco: df

El programa **df** presenta la cantidad de espacio libre, partición por partición (o volumen por volumen). Las unidades/particiones deben estar montadas en orden para obtener esta información. También se puede obtener información NFS de esta manera. A continuación, se da la lista de algunos parámetros de **df**; en la página del manual de este programa se encuentran opciones adicionales (que rara vez se usan).

Opción para df	Descripción
-h	Genera la cantidad de espacio libre en números legibles para las personas, en lugar de en bloques libres.
-l	Sólo da la lista de los sistemas de archivos montados locales. No presenta información acerca de los sistemas de archivos montados de la red.

Para mostrar el espacio libre para todas las unidades localmente montadas, use este comando:

```
[yyang@serverA ~]$ df -l
```

Con el fin de mostrar el espacio libre en un formato legible para las personas, en el sistema de archivos en el cual está ubicado su directorio actual de trabajo, introduzca

```
[yyang@serverA ~]$ df -h .
```

Con el fin de mostrar el espacio libre en un formato legible para las personas, en el sistema de archivos en el cual está ubicado **/tmp**, teclee este comando:

```
[yyang@serverA ~]$ df -h /tmp
```

Sincronizar discos: sync

Como la mayor parte de otros sistemas operativos modernos, Linux mantiene una caché de disco para mejorar la eficiencia. Por supuesto, la desventaja es que no todo lo que usted quiere escrito en el disco habrá de ser escrito en éste en cualquier momento.

Para programar que la caché del disco sea escrita fuera de éste, use el comando **sync**. Si **sync** detecta que la escritura de la caché fuera del disco ya ha sido programada, se instruye al núcleo para que, de inmediato, la vacíe. Este comando no tiene parámetros para la línea de comandos.

Teclee este comando para asegurarse de que se ha vaciado la caché del disco:

```
yyang@serverA ~]$ sync ; sync
```

NOTA La emisión manual de este comando rara vez es necesaria en la actualidad, toda vez que el sistema operativo Linux ejecuta esta labor bastante bien por sí solo.

MOVIMIENTO DE UN USUARIO Y SU DIRECTORIO INICIAL

En esta sección se demostrará cómo reunir algunos de los temas y utilidades cubiertas hasta ahora en este capítulo. El elegante diseño de Linux y UNIX le permite combinar comandos sencillos para realizar operaciones avanzadas.

A veces, en el curso de la administración, podría tener que mover un usuario y sus archivos de una a otra parte. En esta sección, se cubrirá el proceso de mover el directorio inicial de un usuario. En esta sección, va a mover el usuario nombrado “project5” desde su directorio inicial predeterminado `/home/project5` hasta `/export/home/project5`. También tendrá que fijar los permisos y propiedad apropiados de los archivos y directorios del usuario, de modo que éste pueda tener acceso a ellos.

A diferencia de los ejercicios anteriores, que se estuvieron realizando como un usuario común (el usuario yyang), necesitará los privilegios del superusuario para realizar los pasos de este ejercicio.

1. Entre al sistema como raíz y lance una terminal virtual.
2. Cree el usuario que utilizará para este proyecto. El nombre de usuario es “project5.” Teclee

```
[root@serverA ~]# useradd project5
```

3. Use el comando `grep` para ver la entrada correspondiente al usuario que creó, en el archivo `/etc/passwd`. Teclee

```
[root@serverA ~]# grep project5 /etc/passwd
project5:x:502:503::/home/project5:/bin/bash
```

4. Use el comando `ls` para presentar una lista del directorio inicial del usuario. Teclee

```
[root@serverA ~]# ls -al /home/project5
total 56
drwx----- 3 project5 project5 4096 Mar  8 23:50 .
drwxr-xr-x  6 root      root     4096 Mar  8 23:50 ..
-rw-r--r--  1 project5 project5   24 Mar  8 23:50 .bash_logout
-rw-r--r--  1 project5 project5  191 Mar  8 23:50 .bash_profile
-rw-r--r--  1 project5 project5 124 Mar  8 23:50 .bashrc
```

5. Compruebe el espacio total de disco que está utilizando el usuario. Teclee

```
[root@serverA ~]# du -sh /home/project5
76K    /home/project5
```

6. Use el comando `su` para convertirse temporalmente en el usuario. Teclee

```
[root@serverA ~]# su - project5
[project5@serverA ~]$
```

7. Como el usuario project5, vea su directorio actual de trabajo.

```
[project5@serverA ~]$ pwd
/home/project5
```

8. Como usuario de project5, cree algunos archivos vacíos. Teclee

```
[project5@serverA ~]$ touch a b c d e
```
9. Regrese a ser el usuario raíz saliendo del perfil de project5. Teclee

```
[project5@serverA ~]$ exit
```
10. Cree el directorio **/export** que alojará el nuevo directorio inicial del usuario. Teclee

```
[root@serverA ~]# mkdir -p /export
```
11. Ahora use el comando **tar** para archivar y comprimir el directorio inicial actual de project5 (**/home/project5**) y deshaga tar y descomprímalo en su nueva ubicación. Teclee

```
[root@serverA ~]# tar czf - /home/project5 | (cd /export ; tar -xvf -)
```

SUGERENCIA Los guiones (-) que usó aquí con el comando **tar** lo fuerzan a enviar primero su salida y, después, recibir su entrada desde stdin.

12. Use el comando **ls** para asegurarse de que el nuevo directorio inicial se creó de manera apropiada bajo el directorio **/export**. Teclee

```
[root@serverA ~]# ls -R /export/home/
/export/home/:
project5
/export/home/project5:
a b c d e
```
13. Asegúrese de que project5 tenga propiedad completa de todos los archivos y directorios en su nuevo directorio inicial. Teclee

```
[root@serverA ~]# chown -R project5.project5 /export/home/project5/
```
14. Ahora borre el antiguo directorio inicial de project5. Teclee

```
[root@serverA ~]# rm -rf /home/project5
```
15. Bueno, casi hemos terminado. Intente tomar una vez más de manera temporal la identidad de project5. Teclee

```
[root@serverA ~]# su - project5
su: warning: cannot change directory to /home/project5: No such file or directory
-bash-3.00$
```

¡Ah!...dejó de hacer una cosa más. Hemos borrado el directorio inicial del usuario (**/home/project5**), como se especificó en el archivo **/etc/passwd** y eso es por lo que el comando **su** está protestando aquí.

16. Salga del perfil de project5 usando el comando **exit**. Teclee

```
-bash-3.00$ exit
```

17. Ahora usaremos el comando **usermod** para actualizar en forma automática el archivo **/etc/passwd** con el nuevo directorio inicial del usuario. Teclee

```
[root@serverA ~]# usermod -d /export/home/project5 project5
```

18. Use el comando **su** de nuevo para convertirse de manera temporal en project5. Teclee

```
[root@serverA ~]# su - project5  
[project5@serverA ~]$
```

19. Mientras se encuentra dentro como project5, use el comando **pwd** para ver su directorio actual de trabajo. Teclee

```
[project5@serverA ~]$ pwd  
/export/home/project5
```

Esta salida muestra que nuestra migración funcionó bien.

20. Salga del perfil de project5 para convertirse en el usuario raíz y, a continuación, borre el usuario llamado project5 del sistema. Teclee

```
[root@serverA ~]# userdel -r project5
```

Hacer una lista de procesos: ps

El comando **ps** hace una lista de todos los procesos en un sistema, de su estado, tamaño, nombre, propietario, tiempo de CPU, hora del reloj y mucho más. Se dispone de muchos parámetros para la línea de comandos; en la tabla 5-4, se describen los que se usan más a menudo.

Opción para ps

Descripción

-a	Muestra todos los procesos con una terminal controladora, no sólo los procesos actuales del usuario.
-r	Muestra sólo los procesos en ejecución (vea la descripción de estados de los procesos más adelante en esta sección).
-x	Muestra los procesos que no tienen una terminal controladora.
-u	Muestra los propietarios de los procesos.
-f	Presenta las relaciones padre/hijo entre los procesos.
-l	Produce una lista en un formato largo.
-w	Muestra los parámetros de la línea de comandos de un proceso hasta la mitad de una línea.
-ww	Muestra todos los parámetros de la línea de comandos de un proceso sin importar su longitud.

Tabla 5-4. Opciones comunes para ps

El conjunto más común de parámetros que se usa con el comando **ps** es **auxww**. Estos parámetros muestran todos los procesos (sin importar si tienen o no terminal controladora), los propietarios de cada proceso y todos los parámetros de las líneas de comandos. Examinemos la salida muestra de una llamada a **ps auxww**.

```
[yyang@serverA ~]$ ps auxww
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.2	2332	564	?	S	Mar07	0:12	init [3]
root	2	0.0	0.0	0	0	?	SN	Mar07	0:00	[ksoftirqd/0]
root	3	0.0	0.0	0	0	?	S<	Mar07	0:04	[events/0]
root	4	0.0	0.0	0	0	?	S<	Mar07	0:00	[khelper]
root	5	0.0	0.0	0	0	?	S<	Mar07	0:00	[kacpid]
root	1216	0.0	0.2	3004	504	?	S<s	Mar07	0:00	udevd
root	1732	0.0	0.0	0	0	?	S	Mar07	0:00	[kjournald]
root	1733	0.0	0.0	0	0	?	S	Mar07	0:00	[kjournald]
root	1734	0.0	0.0	0	0	?	S	Mar07	0:00	[kjournald]
root	2076	0.0	0.3	3284	584	?	Ss	Mar07	0:02	syslogd -m 0
root	2080	0.0	0.2	1792	468	?	Ss	Mar07	0:00	klogd -x
rpc	2108	0.0	0.3	2216	636	?	Ss	Mar07	0:00	portmap
rpcuser	2128	0.0	0.4	2496	844	?	Ss	Mar07	0:00	rpc.statd
root	2161	0.0	0.3	2972	588	?	Ss	Mar07	0:21	rpc.idmapd
root	2231	0.3	0.2	4348	572	?	Ss	Mar07	5:27	nifd -n
nobody	2261	0.0	0.5	13536	1020	?	Ssl	Mar07	0:00	mDNSResponder
root	2282	0.0	0.2	1600	532	?	Ss	Mar07	0:00	/usr/sbin/acpid
root	2357	0.0	0.8	4248	1532	?	Ss	Mar07	0:00	/usr/sbin/sshd
root	2419	0.0	0.4	5128	828	?	Ss	Mar07	0:01	crond
xfs	2445	0.0	0.7	4528	1524	?	Ss	Mar07	0:00	xfs -droppriv -
daemon	2464	0.0	0.3	2224	640	?	Ss	Mar07	0:00	/usr/sbin/atd
dbus	2474	0.0	0.6	3780	1196	?	Ss	Mar07	0:00	dbus-daemon -1 --
root	2487	0.0	0.5	3436	1032	?	Ss	Mar07	0:00	cups-config-daemon
root	2498	0.0	1.9	5836	3636	?	Ss	Mar07	1:18	hald
root	2530	0.0	0.2	2340	408	tty1	Ss+	Mar07	0:00	/sbin/mingetty tty1
root	2531	0.0	0.2	2620	408	tty2	Ss+	Mar07	0:00	/sbin/mingetty tty2
root	3196	0.0	1.1	9776	2192	?	SNs	Mar07	0:01	cupsd
root	3555	0.0	1.0	7464	2032	?	Ss	Mar07	0:00	sshd: yyang [priv]
yyang	3557	0.0	1.2	7848	2396	?	S	Mar07	0:24	sshd: yyang@pts/0
yyang	3558	0.0	0.7	6128	1444	pts/0	Ss	Mar07	0:06	-bash
yyang	3607	0.0	1.0	7176	2096	?	S	Mar07	0:02	usr/libexec/gconfd-
root	3757	0.0	1.1	7628	2124	?	Ss	Mar08	0:00	sshd: root@pts/2
root	3759	0.0	0.7	4924	1456	pts/2	Ss+	Mar08	0:00	-bash
yyang	3822	0.0	2.9	10500	5616	pts/0	S	Mar08	1:19	xterm
yyang	3824	0.0	0.7	5128	1376	pts/1	Ss	Mar08	0:00	bash
yyang	3842	0.0	0.8	4808	1648	pts/1	S+	Mar08	0:00	ssh 10.0.99.5 -l
root	5272	0.0	0.2	4524	540	?	S	00:37	0:00	sleep 1h
root	5283	0.1	0.6	5184	1152	pts/0	S	00:49	0:00	su - yyang
yyang	5284	0.2	0.7	5684	1368	pts/0	S	00:49	0:00	-bash
yyang	5310	0.0	0.4	4016	772	pts/0	R+	00:50	0:00	ps auxww

La misma primera línea de la salida proporciona los encabezados de las columnas para la lista, como sigue:

- ▼ **USER** Quién posee cuál proceso.
- **PID** Número de identificación del proceso.
- **%CPU** Porcentaje de la CPU tomada por un proceso. Nota: para un sistema con múltiples procesadores, esta columna sumará hasta más del 100%.
- **%MEM** Porcentaje de la memoria tomada por un proceso.
- **VSZ** La cantidad de memoria virtual que está tomando un proceso.
- **RSS** La cantidad de memoria real (residente) que está tomando un proceso.
- **TTY** La terminal controladora de un proceso. Un signo de final de interrogación en esta columna significa que el proceso ya no está conectado a una terminal controladora.
- **STAT** El estado del proceso. Éstos son los estados posibles:
 - ▼ **S** El proceso está durmiendo. Todos los procesos que están listos para ejecutarse (es decir, que tienen tareas múltiples y, en ese momento, la CPU está enfocada en otra parte) estarán dormidos.
 - **R** Proceso que se encuentra en realidad en la CPU.
 - **D** Sueño que no puede interrumpirse (por lo común relacionado con I/O).
 - **T** Proceso que está siendo recorrido por un eliminador de errores o que se ha detenido.
 - ▲ **Z** Proceso que se ha vuelto zombi. Esto significa que 1) el proceso padre no ha reconocido la muerte de su hijo con el uso de la llamada **wait** del sistema, o bien, 2) el padre es **killed** en forma no apropiada, y hasta que ese padre no esté por completo **killed**, el proceso **init** (vea el capítulo 8) no puede anular al propio hijo. Un proceso que se ha convertido en zombi suele indicar un software mal escrito.

Además, la entrada STAT para cada proceso puede tomar uno de los modificadores siguientes: W = Páginas no residentes en la memoria (ésta se ha cambiado por completo); < = Proceso de alta prioridad; N = Tarea de baja prioridad; L = Las páginas de la memoria están bloqueadas allí (lo que suele significar que se necesita de una funcionalidad de tiempo real).

- **START** Fecha en que se inició el proceso.
- **TIME** Cantidad de tiempo que el proceso ha pasado en la CPU.
- ▲ **COMMAND** Nombre del proceso y sus parámetros de la línea de comandos.

Mostrar una lista interactiva de procesos: **top**

El comando **top** es una versión interactiva del **ps**. En lugar de dar una visión estática de lo que está pasando, **top** refresca la pantalla con una lista de los procesos cada dos o tres segundos (ajustable por el usuario). A partir de esta lista, puede volver a establecer las prioridades de los procesos o aplicarles **kill**. En la figura 5-1 se muestra una pantalla de **top**.

PID	USER	PR	NI	VIRT	RES	SHR	S	2CPU	%MEM	TIME+	COMMAND
5311	yyang	17	0	3444	932	1664	R	1.6	0.5	0:00.58	top
2231	root	16	0	4348	572	1368	S	0.7	0.3	5:28.75	nifd
3567	yyang	16	0	7848	2396	3684	S	0.3	1.3	0:24.26	sshd
3822	yyang	16	0	10500	5616	6480	S	0.3	2.9	1:19.59	xterm
1	root	16	0	2332	564	1408	S	0.0	0.3	0:12.80	init
2	root	34	19	0	0	0	S	0.0	0.0	0:00.07	ksoftirqd/0
3	root	5	-10	0	0	0	S	0.0	0.0	0:04.97	events/0
4	root	5	-10	0	0	0	S	0.0	0.0	0:00.02	khelper
5	root	15	-10	0	0	0	S	0.0	0.0	0:00.00	kacpid
18	root	5	-10	0	0	0	S	0.0	0.0	0:00.59	kblockd/0
28	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pdfflush
29	root	15	0	0	0	0	S	0.0	0.0	0:02.68	pdfflush
31	root	12	-10	0	0	0	S	0.0	0.0	0:00.00	aio/0
19	root	15	0	0	0	0	S	0.0	0.0	0:00.00	khabd
30	root	16	0	0	0	0	S	0.0	0.0	0:01.81	ksmpd0
104	root	25	0	0	0	0	S	0.0	0.0	0:00.00	kseriod
178	root	21	0	0	0	0	S	0.0	0.0	0:00.00	scsi_eh_0
185	root	7	-10	0	0	0	S	0.0	0.0	0:00.00	kmirrord/0
196	root	15	0	0	0	0	S	0.0	0.0	0:03.47	kjournald
1216	root	6	-10	3004	504	1340	S	0.0	0.3	0:00.77	udevd
1595	root	6	-10	3528	1008	1680	S	0.0	0.5	0:00.04	dhclient
1732	root	16	0	0	0	0	S	0.0	0.0	0:00.00	kjournald
1733	root	15	0	0	0	0	S	0.0	0.0	0:00.24	kjournald
1734	root	15	0	0	0	0	S	0.0	0.0	0:00.15	kjournald
2037	root	16	0	2068	396	1680	S	0.0	0.5	0:00.02	dhclient

Figura 5-1. Salida de top

La desventaja principal del programa **top** es que se trata de un cerdo para la CPU. En un sistema congestionado, este programa tiende a complicar los aspectos de administración del mismo. Los usuarios inician la ejecución de **top** para ver lo que está pasando, sólo para hallar que algunas otras personas están ejecutando el programa, lo que vuelve incluso más lento el sistema.

De manera predeterminada, **top** se embarca de forma que cualquiera pueda usarlo. Puede ser que encuentre prudente, dependiendo de su entorno, restringir el uso de **top** sólo al raíz. Para hacer esto, como el raíz, cambie los permisos del programa con el comando siguiente:

```
[root@serverA ~]# chmod 0700 `which top`
```

Enviar una señal a un proceso: kill

Este nombre del programa es engañoso: en realidad no anula los procesos. Lo que hace es enviar señales a los procesos en ejecución. De manera predeterminada, el sistema operativo suministra a cada proceso un conjunto estándar de *manejadores de señales* a cada proceso para tratar con las señales entrantes. Desde el punto de vista de un administrador de sistema, la mayor parte de los manejadores comunes son para los números de señales 9 y 15, el proceso anular y terminar, respectivamente. Cuando se llama **kill**, requiere por lo menos un parámetro: el número de identificación del proceso (PID), seguido de obtuvo del comando **ps**. Cuando sólo se pasa el PID, **kill** envía la señal 15. Algunos programas interceptan esta señal y realizan varias acciones, de modo que puedan pararse con limpieza. Otros sólo suspenden su ejecución en sus caminos. De cualquier manera, **kill** no es un método garantizado para hacer que un proceso se detenga.

Señales

Un parámetro opcional del que se dispone para **kill** es **-n**, en donde la **n** representa un número de señal. Como administradores de sistemas, las señales que más nos interesan son la 9 (anular) y la 1 (colgar).

La señal de anular, 9, es la manera maleducada de detener un proceso. En lugar de pedirle a un proceso que se detenga, el sistema operativo sencillamente lo anula. El único momento en que esto fallará es cuando el proceso se encuentra en medio de una llamada del sistema (como una solicitud para abrir un archivo), en cuyo caso el proceso morirá una vez que regrese de esa llamada.

La señal de colgar, 1, es un poco de retroceso hacia los días de la terminal VT100 de UNIX. Cuando se caía la conexión de la terminal de un usuario en medio de una sesión, todos los procesos que se estuvieran ejecutando en esa terminal recibirían una señal de colgar (llamada a menudo SIGHUP o HUP). Esto daba a los procesos una oportunidad para realizar un paro limpio o, en los casos de los procesos en segundo plano, de ignorar la señal. En estos días, se usa una HUP para decirle a ciertas aplicaciones del servidor que vayan a sus archivos de configuración y que los lean (el lector verá esto en acción en varios de los capítulos posteriores). La mayor parte de las aplicaciones sencillamente ignoran la señal.

Aspectos de seguridad

Es obvio que terminar un proceso es una capacidad poderosa, lo que hace importantes las precauciones de seguridad. Los usuarios sólo pueden anular procesos para los que tienen permiso de hacerlo. Si usuarios que no son raíces intentan enviar señales a procesos que no les pertenezcan, se les responde con mensajes de error. El usuario raíz es la excepción para esta limitación; el raíz puede enviar señales a todos los procesos en el sistema. Por supuesto, esto significa que el raíz necesita tener mucho cuidado al usar el comando **kill**.

Ejemplos de uso del comando kill

NOTA Los ejemplos siguientes son muy arbitrarios; los PID que se usan son por completo ficticios y serán diferentes en su sistema.

Use este comando para terminar un proceso con número 205989 de PID.

```
[root@serverA ~]# kill 205989
```

Para tener una anulación casi garantizada del proceso con número 593999, emita el comando

```
[root@serverA ~]# kill -9 593999
```

Teclee lo siguiente para enviar la señal de HUP al programa **init** (lo cual siempre es PID 1):

```
[root@serverA ~]# kill -SIGHUP 1
```

Este comando es igual que teclear

```
[root@serverA ~]# kill -1 1
```

SUGERENCIA ¡Para obtener una lista de todas las señales posibles de las que se dispone, junto con sus equivalentes numéricos, emita el comando `kill -1`!

HERRAMIENTAS DIVERSAS

Las herramientas que siguen no caen en alguna de las categorías específicas que hemos cubierto en este capítulo. Todas colaboran de manera importante a las tareas diarias de la administración de sistemas.

Mostrar el nombre del sistema: `uname`

El programa `uname` produce algunos detalles del sistema que pueden ser de ayuda en varias situaciones. Puede ser que haya usted administrado entradas remotas para una docena de computadoras diferentes ¡y haya perdido la noción de dónde se encuentra! Esta herramienta también resulta de ayuda para los escritores de *scripts*, porque les permite cambiar la trayectoria de un *script* según la información del sistema.

A continuación se dan los parámetros para la línea de comandos de `uname`:

Opción para <code>uname</code>	Descripción
<code>-m</code>	Imprime el tipo de hardware de la máquina (como i686 para Pentium Pro y las mejores arquitecturas).
<code>-n</code>	Imprime el nombre de anfitrión de la máquina.
<code>-r</code>	Imprime el nombre de lanzamiento del sistema operativo.
<code>-s</code>	Imprime el nombre del sistema operativo.
<code>-v</code>	Imprime la versión del sistema operativo.
<code>-a</code>	Imprime todo lo anterior.

Para obtener el nombre del sistema operativo y el de su lanzamiento, haga entrar el comando siguiente:

```
[root@serverA ~] # uname -s -r
```

NOTA La opción `-s` puede parecer desperdiciada (después de todo, sabemos que éste es Linux), pero este parámetro también prueba ser bastante útil en casi todos los sistemas operativos semejantes a UNIX. En una estación de trabajo SGI, `uname -s` dará como respuesta IRIX, o SunOS en una estación de trabajo Sun. Las personas que trabajan en un entorno heterogéneo con frecuencia escriben *scripts* que se comportarán de modo diferente, dependiendo del OS, y `uname` con `-s` una manera uniforme de determinar esa información.

Quién ha entrado: who

En los sistemas que permiten que los usuarios entren a las máquinas de otros usuarios o a servidores especiales, usted querrá saber quién ha entrado. Puede generar un informe de ese tipo mediante el uso del comando **who**:

```
[yyang@serverA ~]$ who
yyang    pts/0          Mar  7 23:18 (10.45.45.2)
yyang    pts/1          Mar  8 01:12
root     pts/2          Mar  8 01:01 (10.45.45.2)
```

Variación de who: w

El comando **w** presenta la misma información que **who** y mucho más en conjunto. Los detalles del informe incluyen quién ha entrado, cuáles son sus terminales, desde dónde han entrado, cuánto tiempo han estado dentro, durante cuánto tiempo han estado ociosos y su utilización de la CPU. La parte superior del informe también le da la misma salida que el comando **uptime**.

```
[yyang@serverA ~]$ w
 01:18:44 up 1 day,  3:43,  3 users,  load average: 0.00, 0.00, 0.00
USER   TTY      FROM           LOGIN@     IDLE     JCPU    PCPU WHAT
yyang   pts/0    10.0.99.2      Mon23     0.00s  1:29    0.06s w
yyang   pts/1    -              Tue01     24:01m  0.53s  0.36s ssh 10.45.45.5 -l
root    pts/2    10.0.99.2      Tue01     1:05    0.56s  0.56s -bash
```

Comutar el usuario: su

Este comando se usó con anterioridad, cuando movimos un usuario y su directorio inicial, y ahora lo discutiremos con brevedad. Una vez que usted ha entrado al sistema como usuario, no necesita salir y regresar para tomar otra identidad (por ejemplo, usuario raíz). En lugar de ello, use el comando **su** para realizar la conmutación. Este comando tiene muy pocos parámetros para la línea de comandos.

Si se ejecuta **su** sin parámetros, de manera automática intentará hacer que usted sea el usuario raíz. Se le pedirá la contraseña del raíz y, si la hace entrar correctamente, caerá hacia un shell raíz. Si ya es el usuario raíz y quiere conmutarse hacia otra ID, no necesita hacer entrar la nueva contraseña cuando use este comando.

Por ejemplo, si ha entrado como el usuario yyang y quiere conmutarse hacia el raíz, teclee este comando:

```
[yyang@serverA ~]$ su
```

Se le pedirá la contraseña del raíz.

Si ya ha entrado como el raíz y quiere conmutarse a, digamos, el usuario yyang, haga entrar este comando:

```
[root@serverA ~]# su yyang
```

No se le pedirá la contraseña de yyang.

El parámetro opcional de guión (-) le dirá a **su** que conmute las identidades y ejecute los *scripts* de entrada para ese usuario. Por ejemplo, si ha entrado como el raíz y quiere comutarse hacia el usuario yyang con todas sus configuraciones de entrada y shell, teclee este comando:

```
[root@serverA ~]# su - yyang
```

EDITORES

Con facilidad, los editores se encuentran entre las más voluminosas de las herramientas comunes, pero también son las más útiles. Sin ellas, hacer cualquier clase de cambio a un texto sería una empresa tremenda. Sin importar cuál sea su distribución de Linux, tendrá unos cuantos editores. Debe tomarse unos cuantos instantes para sentirse cómodo con ellos.

NOTA No todas las distribuciones vienen con todos los editores, cuya lista se da a continuación.

vi

El editor **vi** ha estado por allí en los sistemas basados en UNIX desde la década de 1970 y su interfaz lo hace ver. Se puede argumentar que es uno de los últimos editores en los que en la actualidad se usan un modo de comandos y un modo de entrada de datos separados; como resultado, la mayor parte de los recién llegados encuentra desagradable su uso. Pero antes de que le vuelva la espalda a **vi**, tómese un momento para sentirse cómodo con él. En las situaciones difíciles, puede ser que no cuente con un editor gráfico bonito a su disposición y **vi** es ubicuo en todos los sistemas UNIX.

La versión de **vi** que se embarca con las distribuciones de Linux es **vim** (VI iMproved). En primer lugar, tiene mucho de lo que hizo popular a **vi** y muchas características que lo hacen útil en los entornos de hoy en día (incluyendo una interfaz gráfica, si está ejecutando el X Window System).

Para iniciar **vi**, sencillamente teclee

```
[yyang@serverA ~]$ vi
```

El editor **vim** tiene un tutor en línea que le puede ayudar a iniciarse con rapidez con él. Para lanzar el tutor, teclee

```
[yyang@serverA ~]$ vimtutor
```

Otra manera fácil de aprender más acerca de **vi** es iniciarla y hacer entrar :help. Si no llega jamás a adherirse a **vi**, oprima varias veces la tecla **ESC** y, enseguida, teclee :q! para forzar una salida sin guardar. Si quiere guardar el archivo, teclee :wq.

emacs

Se ha argumentado que **emacs** es todo un sistema operativo por sí mismo. Es grande, rico en características, expansible, programable y asombroso por todas partes. Si está viniendo de un fundamento de GUI, es probable que encuentre a **emacs** como un entorno agradable para trabajar con

él al principio. En su frente, funciona como Notepad (Bloc de notas), en términos de su interfaz. Sin embargo, por debajo, es una interfaz completa para el entorno de desarrollo de GNU, un lector de correo, un lector de noticias, un navegador de la Web e, incluso, para un psiquiatra (bien, no con exactitud).

Para iniciar **emacs**, sencillamente teclee

```
[yyang@serverA ~] $ emacs
```

Una vez que se ha iniciado **emacs**, puede visitar al psiquiatra al presionar **ESC-X** y, a continuación, teclear **doctor**. Para obtener ayuda usando **emacs**, presione **CTRL-H**.

joe

De los editores cuya lista se da aquí, **joe** es el que más semeja a un simple editor de textos. Funciona de manera muy semejante a Notepad y ofrece ayuda en pantalla. Cualquiera que recuerde el conjunto de comandos del WordStar original se sentirá agradablemente sorprendido de ver que todas esas células cerebrales que cuelgan de los comandos **CTRL-K** se pueden poner de regreso para usarse con **joe**.

Para iniciar **joe**, sencillamente teclee

```
[yyang@serverA ~] $ joe
```

pico

El programa **pico** es otro editor inspirado por la sencillez. Por lo común, usado en conjunción con el sistema de lectura de correo Pine, **pico** también se puede usar como un editor único. Como **joe**, puede funcionar de una manera semejante a Notepad, pero **pico** usa su propio conjunto de combinaciones clave. Por fortuna, todas las combinaciones clave de las que se dispone siempre se muestran en la parte de abajo de la pantalla.

Para iniciar **pico**, sencillamente teclee

```
[yyang@serverA ~] $ pico
```



SUGERENCIA El programa **pico** realizará en forma automática cambios de línea. Por ejemplo, si lo está usando para editar archivos de configuración, tenga cuidado de que no haga un cambio de línea para formar dos líneas si, en realidad, debe permanecer como una.

NORMAS

Un argumento que usted escucha con regularidad en contra de Linux es que existen demasiadas distribuciones diferentes y que, al tener múltiples distribuciones, se tiene fragmentación. Llegará el momento en que esta fragmentación conducirá a versiones distintas de Linux que no serán compatibles.

Sin lugar a duda, éste es un completo desatino que se desarrolla sobre “MID” (miedo, incertidumbre y duda). Estos tipos de argumentos suelen provenir de una falsa comprensión del núcleo y de las distribuciones. Sin embargo, la comunidad de Linux se ha dado cuenta que ha crecido pa-

sando la etapa de los entendimientos informales acerca de cómo se deben hacer las cosas. Como resultado, se está trabajando de manera activa en dos normas principales.

La primera es la File Hierarchy Standard (FHS). Éste es un intento por parte de muchas de las distribuciones de Linux de estandarizar una disposición de los directorios, de modo que los desarrolladores tengan momentos fáciles al asegurarse de que sus aplicaciones funcionen a través de múltiples distribuciones, sin dificultad. En la época en que se está escribiendo esto, Red Hat casi ha condescendido por completo y es probable que la mayor parte de las otras distribuciones también lo hagan.

La otra es la Linux Standard Base Specification (LSB). Como la FHS, la LSB es un grupo de normas en las que se especifica qué debe tener una distribución de Linux en términos de bibliotecas y herramientas.

Un desarrollador que supone que una máquina Linux sólo cumple con la LSB y la FHS garantiza una aplicación que funcionará con todas las instalaciones Linux. Todos los distribuidores principales se han unido a estos grupos de normas. Esto debe garantizar que todas las distribuciones para escritorio tendrán una cierta cantidad de bases comunes en las que puede confiar.

Desde el punto de vista de un administrador de sistema, estas normas son interesantes pero no cruciales para administrar una red Linux. Sin embargo, nunca causa daño aprender más acerca de las dos. Para obtener más información acerca de la FHS, vaya a su sitio Web en <http://www.pathname.com/fhs>. Para averiguar más acerca de la LSB, revise <http://www.linuxbase.org>.

RESUMEN

En este capítulo, discutimos la interfaz de línea de comandos de Linux, a través de BASH, muchas herramientas de la línea de comandos y unos cuantos editores. A medida que siga adelante en este libro, encontrará muchas referencias a la información que se da en este capítulo, asegúrese de sentirse cómodo con el trabajo en la línea de comandos. Puede ser que la encuentre un poco irritante al principio, si está acostumbrado a usar una GUI para realizar muchas de las tareas básicas que se mencionan aquí; pero adhiérase a ella. Puede ser que llegue el momento en que sienta usted que trabaja más rápido en la línea de comandos que con la GUI.

Resulta obvio que en este capítulo no se pueden cubrir todas las herramientas de las que usted dispone para la línea de comandos como parte de su instalación predeterminada de Linux. Se recomienda con vehemencia que se tome algo de tiempo para ver algunos de los libros de referencia de los que se dispone. Para obtener un enfoque útil pero no completo para el considerable detalle de los sistemas Linux, consulte la edición más reciente de *Linux in a Nutshell* (varias ediciones para diferentes sistemas, de O'Reilly and Associates). Además, existe una gran cantidad de textos sobre la programación del shell, en varios niveles y desde varios puntos de vista. Adquiera cualquiera que le convenga; la programación del shell es una habilidad que vale la pena aprender, incluso si usted no realiza la administración de un sistema.

Y por encima de todo lo demás, L. E. B. M.; es decir, lea el *bonito* manual (páginas man).

CAPÍTULO 6



Inicialización
y apagado

Conforme los sistemas operativos se han vuelto más complejos, el proceso de arrancar y parar se han vuelto más detallados. Cualquiera que ha pasado por la transición de un sistema directo basado en DOS hacia uno basado en Windows 2003/XP ha experimentado esta transición de primera mano. No sólo es el sistema operativo central que se desarrolló y se hizo parar, sino también es una lista impresionante de servicios que deban arrancarse y detenerse. Como Windows, Linux comprende una lista impresionante de servicios que se ponen en acción como parte del procedimiento de arranque.

En este capítulo discutimos la inicialización del sistema operativo Linux con GRUB y LILO. A continuación avanzaremos a través de los procesos de arranque y paro del entorno de Linux. Discutimos los scripts que automatizan este proceso así como las partes de éste para las cuales es aceptable la modificación.

NOTA Aplique una dosis liberal de sentido común al seguir los ejercicios prácticos de este capítulo en un sistema real. Conforme experimente con la modificación de los scripts de arranque y paro, tenga presente que es posible que lleve su sistema hacia un estado no funcional que no se puede recuperar con el reinicio. No se meta con un sistema de producción y, si debe hacerlo, primero asegúrese de que respalda todos los archivos que desea cambiar, y, lo que es más importante, tenga listo un disco de arranque (o algún otro medio de inicialización) que le pueda ayudar a recuperar.

CARGADORES DE INICIALIZACIÓN

Con cualquier sistema operativo, para arrancar el hardware de una PC estándar necesita lo que se llama un *cargador de inicialización*. Si sólo ha tratado con Windows en una PC, es probable que nunca haya necesitado interactuar en forma directa con uno de estos cargadores. El cargador de inicialización es el primer programa de software que se ejecuta cuando arranca una computadora; es el responsable de entregar el control del sistema al sistema operativo.

Por lo general, el cargador de inicialización residirá en el Master Boot Record (MBR) del disco y sabe cómo levantar el sistema operativo y hacer que se ejecute. PartitionMagic viene con una herramienta llamada Boot Magic que soporta Linux. Las elecciones principales que vienen con las distribuciones de Linux son GRUB (el Grand Unified Bootloader) y LILO (Linux Loader). Cubriremos principalmente GRUB porque es el cargador de inicialización más común que se embarca con las distribuciones más recientes de Linux y también porque tiene muchas más características que LILO. Por razones históricas, sólo se hace una breve mención de LILO. Tanto LILO como GRUB se puede configurar para inicializar otros sistemas operativos no nativos.

GRUB

En las distribuciones más modernas de Linux, se usa GRUB como el cargador de inicialización predeterminado en el transcurso de la instalación. GRUB es el cargador predeterminado para Fedora Core, RHEL, SuSE, Mandrake y una multitud de otras distribuciones de Linux. La misión de GRUB es cumplir con la Multi-boot Specification y ofrece muchas características.

NOTA El lector podría advertir que GRUB es un lanzamiento de software previo al 1.0, también conocido como software alfa. No se asuste por esto. Considerando que los vendedores principales de Linux lo usan en sus instalaciones, es probable que sea un código de calidad “alfa”. Sólo se le están haciendo al código eliminación de errores. La versión estable de GRUB también se conoce como GRUB Legacy. GRUB 2 está pasando a ser la generación siguiente de GRUB.

El proceso de inicialización de GRUB se lleva a cabo por etapas. Archivos imagen de GRUB se encargan de cada etapa, en tal forma que cada etapa sirve de ayuda a la que le sigue. Dos de las etapas son esenciales y cualquiera de las otras son opcionales y dependen de la estructura particular del sistema.

Etapa 1

El archivo imagen que se usa en esta etapa es esencial y se usa para que, en primer lugar, arranque GRUB. Por lo común está incrustado en el MBR de un disco o en el sector de arranque de una partición. Al archivo que se usa en esta etapa se le da, con toda propiedad, el nombre de **stage1**. Una imagen de la etapa 1 puede estar próxima a la etapa 1.5 de carga o directamente a la 2.

Etapa 2

Las imágenes de la etapa 2 en realidad constan de dos tipos de imágenes: el archivo imagen intermedio (imagen opcional) y el **stage2** real. Para desdibujar todavía más las cosas, a las imágenesopcionales se les llama etapa 1.5. Las imágenes de esta etapa sirven como un puente entre la 1 y la 2. Las imágenes de la etapa 1.5 constituyen un sistema específico de archivos; es decir, entienden la semántica de uno de los sistemas de archivos o del otro.

Las imágenes de la etapa 1.5 tienen nombres de la forma *x*_etapa_1_5, en donde *x* puede ser un sistema de archivos del tipo e2fs, reiserfs, fat, jfs, minix, xfs, etcétera. Por ejemplo, la imagen de la etapa 1.5 que se requerirá para cargar un OS que reside en un sistema de archivos FAT tendrá un nombre como **fat_stage1_5**. Las imágenes de la etapa 1.5 permiten a GRUB tener acceso a varios sistemas de archivos. Cuando se usa, la imagen de la etapa 1.5 ayuda a ubicar la imagen de la etapa 2 como un archivo dentro del sistema de archivos.

A continuación viene la imagen **stage2** real. Es la parte central de GRUB. Contiene el código real para cargar el núcleo que inicializa el OS, presenta el menú de inicialización y también contiene el shell de GRUB desde el cual se pueden introducir los comandos de éste. El shell de GRUB es interactivo y ayuda a hacer que este último sea muy flexible. Por ejemplo, se puede usar el shell para arrancar elementos que no se encuentran en la lista en curso del menú de inicialización de GRUB o para inicializar el OS a partir de un medio alterno soportado.

Otros tipos de imágenes de la etapa 2 son la **stage2_eltorito**, **nbgrub** y la **pxegrub**. La imagen stage2_eltorito es una de arranque para CD-ROM. Tanto la imagen nbgrub como la pxegrub son imágenes de arranque del tipo de red que se pueden usar para inicializar un sistema sobre la red (usando BOOTP, DHCP, PXE, Etherboot o algo semejante). Una lista rápida del contenido del directorio **/boot/grub** de la mayor parte de las distribuciones de Linux mostrará algunas de las imágenes de GRUB.

Convenciones usadas en GRUB

En GRUB se tiene una manera propia especial de hacer referencia a los dispositivos (unidades de CD-ROM, unidades de disquetes, unidades de disco duro, etcétera). El nombre del dispositivo

tiene que estar encerrado entre paréntesis, “()”. En GRUB se empieza a numerar sus dispositivos y particiones desde cero, *no* desde uno. Por lo tanto, en GRUB se haría referencia a la unidad de disco duro master IDE, en el controlador primario IDE como (hd0), en donde “hd” significa unidad de “disco duro” y el número cero significa que es el master IDE primario.

Con el mismo sentido, en GRUB se hará referencia a la tercera partición en el cuarto disco duro (es decir, el esclavo en el controlador IDE secundario) como “(hd3,3). En GRUB, para referirse al disquete completo se diría “(fd0)”; en donde por “fd” se quiere dar a entender “disquete (disco flexible)”.

Instalación de GRUB

En la mayor parte de las distribuciones de Linux, se le dará a elegir que se instale y configure el cargador de inicialización en el inicio de la instalación del sistema operativo. Por tanto, lo normal sería que no necesitará instalar en forma manual GRUB durante el uso normal del sistema.

Sin embargo, hay ocasiones, por accidente o por diseño, en que usted no cuenta con un cargador de inicialización. Podría ser accidental si, por ejemplo, sin desearlo sobreescrbiera en su sector de inicialización o si, de la misma manera, otro sistema operativo eliminara GRUB. Podría ser por diseño si, por ejemplo, quiere estructurar su sistema para la inicialización dual con otro sistema operativo (Windows u otra distribución de Linux).

En esta sección, le haremos recorrer los pasos para tener GRUB instalado (o reinstalado) en su sistema. Existen varias maneras en que se puede lograr esto. Puede realizarlo de la manera fácil desde el interior del OS en ejecución, usando la utilidad **grub-install**, o bien, usando la interfaz nativa de línea de comandos de GRUB. Puede obtener esta interfaz usando lo que se llama disquete de inicialización de GRUB, empleado un CD de inicialización de GRUB o desde un sistema que tenga instalado el software de éste.

NOTA GRUB sólo se instala una vez. Cualesquier modificaciones se almacenan en un archivo de texto, y no es necesario escribir cualesquier cambios cada vez en el MBR o en el sector de inicialización de la partición.

Respaldo del MBR

Antes de seguir adelante con los ejercicios que siguen, es una buena idea hacer un respaldo de su actual MBR, “conocido como bueno”. Es fácil hacer esto usando el comando **dd**. Puesto que el MBR del disco duro de una PC reside en los primeros 512 bytes del disco, puede copiar con facilidad esos 512 bytes en un archivo (o en un disquete) al teclear

```
[root@serverA ~]# dd if=/dev/hda of=/tmp/COPY_OF_MBR bs=512 count=1
1+0 records in
1+0 records out
```

Este comando guardará el MBR en un archivo llamado **COPY_OF_MBR**, bajo el directorio **/tmp**.

Creación de un CD de Inicialización/Rescate

Otra medida precautoria que hay que tomar antes de realizar cualquier operación que conduzca a un sistema que no se pueda reiniciar es crear un CD de rescate. Entonces se usa este CD para

inicializar el sistema en el caso de accidentes (el CD también se puede usar para otros fines y siempre debe guardarse a mano).

El CD de inicialización es muy específico de un sistema y se estructura de manera automática a partir de la información actual que se extrae de su sistema. Usará el comando **mkbootdisk** para generar una imagen ISO que se puede quemar en un CD-ROM en blanco. Para generar una imagen ISO nombrada **BOOT-CD.iso** para su núcleo en ejecución y guardar el archivo imagen bajo el directorio **/tmp**, teclee

```
[root@serverA ~]# mkbootdisk --device /tmp/BOOT-CD.iso --iso `uname -r`
```

A continuación, necesitará hallar una manera de quemar/escribir la imagen de CD creada en un CD en blanco. Si tiene un quemador de CD instalado en la caja de Linux, puede usar la utilidad **cdrecord** para lograr esto, al emitir el comando

```
[root@serverA ~]# cdrecord speed=4 -eject --dev=/dev/hdc /tmp/BOOT-CD.iso
```

Entonces debe poner fecha al disco y rotularlo según un nombre descriptivo.

SUGERENCIA También se puede usar la utilidad **mkbootdisk** para crear un disquete de inicialización. Pero debido a las diferencias entre los núcleos de Linux en la serie de la versión 2.4 y en la de la versión 2.6, ya no resulta muy directo crear un disco de inicialización que se ajustará al espacio limitado (1.44MB) que ofrece un disquete. Si tiene su sistema para satisfacer las restricciones de tamaño, todo lo que necesita hacer para crear un disquete de inicialización es insertar un disquete en blanco en la unidad y emitir el comando

```
[root@serverA ~]# mkbootdisk --device /dev/fd0 `uname -r`
```

Instalación de GRUB desde el shell del mismo

Ahora que hemos tratado las medidas de seguridad, podemos proceder a examinar GRUB por completo. En esta sección, aprenderá cómo instalar GRUB en forma nativa, usando el shell de comandos del propio GRUB desde el interior del sistema operativo de Linux en ejecución. Normalmente seguirá esta ruta si, por ejemplo, en la actualidad tiene otro tipo de cargador de inicialización (como LILO o el NT Loader, ntldr) pero desea reemplazar ese cargador con GRUB o sobrescribir éste sobre aquel.

1. Lance el shell de GRUB emitiendo el comando **grub**. Teclee

```
[root@serverA ~]# grub
GNU GRUB version 0.95 (640K lower / 3072K upper memory)
[ Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename.]
grub>
```

2. Presente el dispositivo actual de GRUB. Teclee

```
grub> root
(fd0): Filesystem type unknown, partition type 0x8e
```

La salida muestra que GRUB usará de modo predeterminado la primera unidad de disquete (fd0) como su dispositivo raíz, a menos que usted le diga otra cosa.

3. Fije el dispositivo raíz de GRUB a la partición que contiene el directorio de inicialización en el disco duro local. Teclee

```
grub> root (hd0,0)
Filesystem type is ext2fs, partition type 0x83
```

NOTA El directorio de inicialización puede estar o no en la misma partición que aloja el directorio raíz (/). En el curso de la instalación del OS en nuestro sistema muestra, el directorio **/boot** se almacenó en la partición /dev/hda1 y, de donde, usamos el dispositivo (hd0,0), de GRUB.

4. Asegúrese de que se puede hallar la imagen **stage1** en el dispositivo raíz. Teclee

```
grub> find /grub/stage1
(hd0,0)
```

La salida quiere decir que la imagen **stage1** estaba ubicada en el dispositivo (hd0,0).

5. Por último, instale el cargador de inicialización GRUB directamente en el MBR del disco duro. Teclee

```
grub> setup (hd0)
Checking if "/boot/grub/stage1" exists... no
Checking if "/grub/stage1" exists... yes
Checking if "/grub/stage2" exists... yes
Checking if "/grub/e2fs_stage1_5" exists... yes
Running "embed /grub/e2fs_stage1_5 (hd0)"... 16 sectors are embedded.
succeeded
Running "install /grub/stage1 (hd0) (hd0)1+16 p (hd0,0)/grub/stage2
/grub/grub.conf"... succeeded
Done.
```

6. Abandone el shell de GRUB. Teclee

```
grub> quit
```

Ha terminado. Pero debe observar que, en realidad, no hizo cambios serios al sistema, porque sencillamente reinstaló GRUB en el MBR (en donde se usa que esté). Lo normal es que, en este punto, reinicie la máquina para asegurarse que todo está funcionando como debe ser.

SUGERENCIA Un script muy sencillo de usar que le puede ayudar a realizar todos los pasos detallados en el ejercicio anterior, con un solo comando, es el script **grub-install** (vea la página man grub-install). Este método no siempre es perfecto y los autores del software GRUB admiten que es una ruta menos segura para tomar. Sin embargo, casi siempre funciona bien.

El disquete de inicialización de GRUB

Creemos un disquete de GRUB. Esto le permitirá inicializar el sistema usando el disquete y usar GRUB para escribirse o instalarse por sí mismo en el MBR. Esto resulta útil en especial si su sistema no cuenta en la actualidad con un cargador de inicialización instalado pero usted tiene acceso a otro sistema que tiene GRUB instalado.

La idea general de usar un disquete de inicialización de GRUB es que se supone que, en la actualidad, usted tiene un sistema con un cargador de inicialización que no se puede arrancar, o está corrupto o no lo desea, y ya que el sistema no se puede inicializar por sí mismo desde el disco duro, necesita otro medio con el cual arrancar dicho sistema. Y, de este modo, puede usar un disquete o un CD de GRUB. Usted desea cualquier medio mediante el cual pueda tener acceso al shell de GRUB, de modo que pueda instalar este último en el MBR y, a continuación, inicializar el OS.

En primer lugar, necesita localizar las imágenes de GRUB, ubicadas de modo predeterminado en el directorio `/usr/share/grub/i386-redhat/` en un sistema Fedora Core (en SuSE se almacenan los archivos imágenes de GRUB en el directorio `/usr/lib/grub/`).

Use el comando `dd` para escribir las imágenes de **stage1** y **stage2** en el disquete.

1. Cambie al directorio que contiene las imágenes de GRUB en su sistema. Teclee

```
[root@serverA ~]# cd /usr/share/grub/i386-redhat/
```

2. Escriba el archivo **stage1** en los primeros 512 bytes del disquete. Teclee

```
[root@serverA i386-redhat]# dd if=stage1 of=/dev/fd0 bs=512 count=1  
1+0 records in  
1+0 records out
```

3. Escriba la imagen **stage2** inmediatamente después de la primera imagen. Teclee

```
[root@serverA i386-redhat]# dd if=stage2 of=/dev/fd0 bs=512 seek=1  
202+1 records in  
202+1 records out
```

SUGERENCIA También puede usar el comando `cat` para hacer lo mismo que se hizo en los dos últimos pasos de una sola vez. El comando para hacer esto será

```
[root@serverA i386-redhat]# cat stage1 stage2 > /dev/fd0.
```

Su disquete de GRUB está ahora listo. Ahora puede lanzarlo de este disquete de modo que pueda instalar el cargador de inicialización GRUB.

Instalación de GRUB en el MBR usando un disquete de él

Asegúrese que el disquete de GRUB que creó se encuentra en la unidad de disquete. Reinicie el sistema y use el disquete como su medio de inicialización (si es necesario, ajuste los valores del BIOS). Después que el sistema se ha inicializado a partir del disquete de GRUB, se le presentará un mensaje de `grub>`.

Fije el dispositivo raíz para GRUB en la partición de inicialización de usted (o en la partición que contiene el directorio `/boot`). En nuestro sistema de ejemplo, el directorio `/boot` reside en la partición `/dev/hda1` (`hd0,0`). Para hacer esto, teclee el comando siguiente:

```
grub> root (hd0,0)
```

Ahora puede escribir GRUB en el MBR, usando el comando `setup`:

```
grub> setup (hd0,0)
```

Eso es todo, ha terminado. Ahora puede reiniciar el sistema *sin* el disquete de GRUB. Ésta es una buena manera de hacer que GRUB recupere la administración del MBR, si con anterioridad otro administrador de inicialización hubiera sobrescrito en él.

Configuración de GRUB

Ya que sólo tiene que instalar GRUB una vez en el MBR o en la partición que prefiera, se puede usted dar el lujo de sencillamente editar un archivo de texto, **/boot/grub/menu.1st**. Cuando haya terminado de editar este archivo, puede de manera franca reiniciar y seleccionar el núcleo nuevo que agregó a la configuración. El archivo de configuración se observa como lo que se da a continuación (por favor, vea que se han agregado a la salida los números de líneas 1 a 16, para facilitar la lectura):

```
[root@serverA ~]# cat /etc/grub.conf
1) # grub.conf generated by anaconda
2) # Note that you do not have to rerun grub after making changes to this file
3) # NOTICE: You have a /boot partition. This means that
4) #           all kernel and initrd paths are relative to /boot/, eg.
5) #           root (hd0,0)
6) #           kernel /vmlinuz-version ro root=/dev/VolGroup00/LogVol00
7) #           initrd /initrd-version.img
8) #boot=/dev/had
9) default=0
10) timeout=5
11) splashimage=(hd0,0)/grub/splash.xpm.gz
12) hiddenmenu
13) title Fedora Core (2.6.11-1.1369_FC4)
14) root (hd0,0)
15) kernel /vmlinuz-2.6.11-1.1369_FC4 ro root=/dev/VolGroup00/LogVol00
16) initrd /initrd-2.6.11-1.1369_FC4.img
```

Enseguida, se discuten las entradas en el archivo de configuración para GRUB que acaba de darse:

- ▼ Líneas 1-8 Todas las líneas que empiezan con el signo de número (#) son comentarios y se ignoran.
- Línea 9, **default** Esta directiva le dice a GRUB cuál entrada arrancar en forma automática. La numeración se inicia desde cero. El archivo muestra antes dado sólo contiene una entrada: la titulada “Fedora Core (2.6.11-1.1369_FC4)”.
- Línea 10, **timeout** Esto significa que GRUB arrancará en forma automática la entrada predeterminada después de transcurridos cinco segundos. Se puede interrumpir esto al presionar cualquier botón del teclado antes de que el contador se detenga.
- Línea 11, **splashimage** En esta línea se especifica el nombre y ubicación de un archivo de imagen que tiene que ser presentado en el menú de inicialización. Éste es opcional y puede ser cualquier imagen personalizada que se ajuste a las especificaciones de GRUB.

- Línea 12, **hiddenmenu** Con esta entrada se esconde el menú usual de GRUB. Es una entrada opcional.
- Línea 13, **title** Ésta se usa para presentar un título o descripción cortos para las entradas siguientes que define. El campo de título marca el principio de una nueva entrada de inicialización en GRUB.
- Línea 14, **root** El lector debe advertir, con base en la lista precedente, que GRUB todavía mantiene su convención de nombramiento de los dispositivos [por ejemplo, (hd0,0), en lugar del /dev/hda1 usual de Linux].
- Línea 15, **kernel** Usada para especificar la trayectoria hacia una imagen del núcleo. El primer argumento es la trayectoria a la imagen del núcleo en una partición. Cualesquiera otros argumentos se pasan al núcleo como parámetros de la inicialización.

Note que los nombres de las trayectorias son relativos al directorio **/boot**; así entonces, por ejemplo, en lugar de especificar que la trayectoria hacia el núcleo es “`/boot/vmlinuz-2.6.11-1.1369_FC4`”, el archivo de configuración de GRUB hace referencia a esta trayectoria como “`vmlinuz-2.6.11-1.1369_FC4`”.

- ▲ Línea 16, **initrd** La opción **initrd** le permite a usted cargar los módulos del núcleo desde una imagen, no los módulos desde **/lib/modules**. Vea las páginas info de GRUB, que se encuentran a través del comando **info**, para obtener más información sobre las opciones de configuración.

NOTA El lector podría preguntarse para qué es en realidad la opción **initrd**. Básicamente, ésta permite a la distribución usar un núcleo genérico que sólo soporta el sistema nativo de archivos de Linux, el cual en la actualidad es el ext3. El problema que se presenta es que usted podría necesitar un módulo de sistema de archivos para cargar todos sus módulos nuevos; si, por ejemplo, prefiere instalar el sistema de archivos ReiserFS. Éste es el problema de la gallina y el huevo; es decir, cuál llegó primero. La solución es proporcionar el núcleo con una imagen que contenga los módulos necesarios que se puedan cargar para tener el resto de los módulos.

Adición de un núcleo nuevo para inicializar con GRUB

En esta sección, aprenderá cómo agregar en forma manual una nueva entrada de inicialización al archivo de configuración de GRUB. Si está compilando e instalando un núcleo nuevo a mano, necesitará hacer esto de modo que pueda arrancar con él para probarlo o usarlo. Si, por otra parte, está instalando o modernizando el núcleo de Linux con el uso de un RPM preempacado, esto suele llevarse a cabo en forma automática para usted.

Debido a que usted no tiene un núcleo nuevo de Linux para instalar en el sistema, en este ejercicio sólo agregará una entrada ficticia al archivo de configuración de GRUB. La nueva entrada nada hará que sea de utilidad; sólo se está haciendo para fines de ilustración.

Enseguida se da un resumen de lo que learemos recorrer:

Hará una copia del núcleo predeterminado actual que su sistema usa y la llamaremos copia **duplicate-kernel**. También hará una copia de la imagen **initrd** correspondiente para el núcleo y

la nombrará **duplicate-initrd**. A continuación, creará una entrada para el supuestamente núcleo nuevo y le dará un título descriptivo como “**The Duplicate Kernel**”.

Además de la entrada de inicialización anterior, también creará otra entrada que no haga más que cambiar los colores de primero y segundo planos del menú de inicialización de GRUB. Empecemos:

1. Cambie su directorio actual de trabajo hacia el directorio **/boot**. Teclee

```
[root@serverA ~]# cd /boot
```
2. Haga una copia de su núcleo actual y nombre esa copia como **duplicate-kernel**. Teclee

```
[root@serverA ~]# cp vmlinuz-2.6.11-1.1369_FC4 duplicate-kernel
```
3. Haga una copia de la imagen de **initrd** y déle el nombre de **duplicate-initrd**. Teclee

```
[root@serverA ~]# cp initrd-2.6.11-1.1369_FC4.img duplicate-initrd.img
```
4. Cree una entrada para los nuevos seudonúcleos en el archivo de configuración **/boot/grub/menu.1st**, usando cualquier editor de textos con el que se sienta cómodo (en este ejemplo, se usa el editor **vim**). Teclee el texto siguiente al final del archivo:

```
title The Duplicate Kernel
color yellow/black
root (hd0,0)
kernel /duplicate-kernel ro root=/dev/VolGroup00/LogVol00
initrd /duplicate-initrd.img
```

5. Cree otra entrada que cambiará los colores de primero y segundo planos del menú, cuando se seleccione. Los colores del menú se cambiarán hacia amarillo y negro cuando se seleccione esta entrada. Introduzca el texto que sigue al final del archivo (debajo de la entrada que creó en el paso anterior):

```
title The change color entry
color yellow/black
```

6. Haga un comentario acerca de la entrada **splashimage**, en la parte de arriba del archivo. La presencia de la imagen llamativa impedirá que sus nuevos colores personalizados de primero y segundo planos se presenten de manera apropiada. La entrada de comentario para la imagen llamativa se verá como esto:

```
#splashimage=(hd0,0)/grub/splash.xpm.gz
```

7. Por último, haga un comentario acerca de la entrada **hiddenmenu** del archivo, de modo que el menú Boot aparecerá mostrando las nuevas entradas de usted, en lugar de ser escondidas. La entrada de comentario debe mirarse como

```
#hiddenmenu
```

8. Guarde los cambios que ha hecho en el archivo y reinicie el sistema.

El archivo **/boot/grub/menu.1st** final (con algunos de los campos de comentarios eliminados) se semejará al que se muestra enseguida:

```
[root@serverA boot]# cat /boot/grub/menu.1st
# grub.conf generated by anaconda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Fedora Core (2.6.11-1.1369_FC4)
    root (hd0,0)
    kernel /vmlinuz-2.6.11-1.1369_FC4 ro root=/dev/VolGroup00/LogVol00
    initrd /initrd-2.6.11-1.1369_FC4.img
title The Duplicate Kernel
    color yellow/black
    root (hd0,0)
    kernel /duplicate-kernel ro root=/dev/VolGroup00/LogVol00
    initrd /duplicate-initrd.img
title The change color entry
    color yellow/black
```

Cuando se reinicia el sistema, debe ver un menú de GRUB semejante al que se muestra enseguida:



9. Después de que aparece el menú de GRUB, seleccione “The change color entry” y oprima ENTER. El color del menú debe cambiar al que especificó en el archivo **menu.1st**, usando la directiva de color.
10. Por último, verifique que es capaz de inicializar la entrada del núcleo nuevo que creó, es decir, la “duplicate kernel entry”. Seleccione la entrada para “The Duplicate Kernel” y, a continuación, presione ENTER.

LILO

LILO, abreviatura de Linux Loader, es un administrador de inicialización que le permite inicializar múltiples sistemas operativos, siempre que cada sistema exista en su propia partición (bajo los sistemas basados en la PC, también debe existir la partición *completa* de inicialización debajo de la frontera del cilindro 1024). Además de arrancar múltiples sistemas operativos, con LILO puede elegir varias configuraciones o versiones del núcleo para arrancar. Esto resulta práctico en especial cuando está tratando con modernizaciones del núcleo, antes de adoptarlas.

La gran imagen con LILO es directa: un archivo de configuración (**/etc/lilo.conf**) especifica cuáles particiones se pueden arrancar y, si una de ellas es Linux, cuál núcleo cargar. Cuando se ejecuta el programa/**sbin/lilo**, toma esta información de la partición y reescribe el sector de inicialización con el código necesario para presentar las opciones según se especifican en el archivo de configuración. En el momento del arranque, se presenta un mensaje (por lo común, **lilo:**) y usted tiene la opción de especificar el sistema operativo (en general, se puede seleccionar un pre-determinado después de un intervalo). LILO carga el código necesario, el núcleo, de la partición seleccionada y le pasa todo el control a él.

LILO es lo que se conoce como un cargador de inicialización en dos etapas. En la primera, se carga el propio LILO en la memoria y le pide a usted que arranque las instrucciones con el mensaje **“lilo:”** o con un menú a colores de inicialización. Una vez que usted selecciona el OS que se va a arrancar y oprime ENTER, LILO hace entrar la segunda etapa, la inicialización del sistema operativo Linux.

Como se dijo al principio en el capítulo, LILO ha dejado un tanto de ser favorecido con la mayor parte de las distribuciones más recientes de Linux. ¡En algunas de las distribuciones incluso no le dan la posibilidad de seleccionar o elegir LILO como su administrador de inicialización!

SUGERENCIA Si está familiarizado con el proceso de arranque de MS Windows, puede concebir a LILO como comparable al cargador de OS (NTLDR). De manera análoga, el archivo de configuración de LILO, **/etc/lilo.conf**, es comparable al **BOOT.INI** (lo cual es típico que esté escondido).

Arranque

En esta sección, supondré que ya está familiarizado con el proceso de inicialización de otros sistemas operativos y, de este modo, ya conoce el ciclo de arranque de su hardware. En esta sección, se cubrirá el proceso de arranque del sistema operativo. Empezaremos con el cargador de inicialización de Linux (por lo común, GRUB para las PC).

Carga del núcleo

Una vez que se ha iniciado GRUB y que usted ha seleccionado Linux como el sistema operativo que se va a arrancar, lo primerísimo que tiene que estar cargado es el núcleo. Tenga presente que, en este punto, ningún sistema operativo existe en la memoria, y las PC (por su diseño desafortunado) no tienen manera fácil de obtener acceso a toda su memoria. Por consiguiente, debe cargarse el núcleo por completo en el primer megabyte de la RAM de la que se dispone. Para realizar esto, el núcleo se comprime. La cabeza del archivo contiene el código necesario para llevar a la CPU a un modo protegido (eliminando de este modo la restricción de memoria) y descomprime el resto del núcleo.

Ejecución del núcleo

Con el núcleo en la memoria, se puede empezar a ejecutar. Sólo conoce cualquiera que sea la funcionalidad integrada en él, lo cual significa que, en este punto, cualesquier partes del núcleo compiladas como módulos son inútiles. En el preciso mínimo, el núcleo debe tener código suficiente como para estructurar su subsistema virtual de memoria y el sistema raíz de archivos (por lo común, el sistema de archivos ext3). Una vez que ha arrancado el núcleo, una sonda del hardware determina cuáles controladores de dispositivos deben inicializarse. A partir de aquí, el núcleo puede *montar* el sistema raíz de archivos (el lector puede encontrar un paralelismo entre este proceso con el que Windows sea capaz de reconocer y tener acceso a su unidad C). El núcleo monta el sistema raíz de archivos e inicia un programa llamado **init**, el cual se discute en la sección siguiente.

EL PROCESO INIT

El proceso **init** es el primero que no es del núcleo que se arranca y, como consecuencia, siempre lleva el número ID de proceso de 1. **init** lee su archivo de configuración **/etc/inittab** y determina el *nivel de ejecución* en donde debe arrancar. En esencia, un nivel de ejecución dicta el comportamiento del sistema. Cada nivel (designado por entero entre 0 y 6) sirve para un fin específico. Si existe, se selecciona un nivel de ejecución de **initdefault**; de lo contrario, se le pide a usted que proporcione un valor de nivel de ejecución.

Los valores del nivel de ejecución son como sigue:

- | | |
|---|--------------------------------------------------------------------------------------------------------------------|
| 0 | Detener el sistema |
| 1 | Introducir el modo de un solo usuario (no se activa el trabajo en red) |
| 2 | Modo de usuarios múltiples, pero sin NFS |
| 3 | Modo completo de usuarios múltiples (operación normal) |
| 4 | Sin usar |
| 5 | Igual al nivel de ejecución 3, excepto que usando una conexión de X Window System, en lugar de una basada en texto |
| 6 | Reiniciar el sistema |

Cuando se dice introducir un nivel de ejecución, **init** ejecuta un script según lo dicta el archivo **/etc/inittab**. El nivel predeterminado de ejecución en el que el sistema se inicializa lo determina la entrada **initdefault** que se encuentra en el archivo **/etc/inittab**. Si, por ejemplo, la entrada en el archivo es

```
id:3:initdefault:
```

esto significa que el sistema se inicializará en el nivel de ejecución 3. Pero si, de lo contrario, la entrada en el archivo es

```
id:5:initdefault:
```

esto significa que el sistema se inicializará en el nivel de ejecución 5, ejecutándose el subsistema X Window con una pantalla de conexión gráfica.

SCRIPTS RC

En la sección anterior se mencionó que el archivo **/etc/inittab** especifica cuáles scripts se van a ejecutar cuando se cambie el nivel de ejecución. Estos scripts son responsables de arrancar o de detener los servicios que son particulares del nivel.

Debido al número de servicios que necesitan ser administrados, se usan los scripts **rc**. El principal, **/etc/rc.d/rc**, es el responsable de llamar los scripts apropiados, en el orden correcto, para cada nivel de ejecución. Como el lector puede imaginar, ¡un script como ése con facilidad podría volverse en extremo incontrolable! Para evitar que esto suceda, se usa un sistema un poco más elaborado.

Para cada nivel de ejecución, existe un subdirectorio en el directorio **/etc/rc.d**. Estos subdirectorios del nivel de ejecución siguen el esquema de nombramiento de **rcX.d**, en donde X es el nivel de ejecución. Por ejemplo, todos los scripts para el nivel de ejecución 3 están en **/etc/rc.d/rc3.d**.

En los directorios de los niveles de ejecución, los vínculos simbólicos se hacen para los scripts en el directorio **/etc/rc.d/init.d**. Sin embargo, en lugar de usar el nombre del script como existe en este directorio, a los enlaces simbólicos se les asigna un prefijo con **S**, si ese script es para iniciar un servicio, o **K**, si el script es para detener (o anular) un servicio. Note que estas dos letras son sensibles a las mayúsculas y minúsculas. Debe usar mayúsculas o los scripts de arranque no las reconocerán.

En muchos casos, el orden en el cual se ejecutan estos scripts establece una diferencia. (Por ejemplo, usted no puede iniciar los servicios que se apoyen en una interfaz configurada de red, ¡sin primero activar y configurar la interfaz de red!) Para hacer cumplir el orden se le agrega un sufijo a la **S** o a la **K**. Los números menores se ejecutan antes que los mayores; por ejemplo, **/etc/rc.d/rc3.d/S10network** se ejecutan antes que **/etc/rc.d/rc3.d/S55sshd** (**S10network** configura los ajustes de la red y **S55sshd** inicia el servidor SSH).

Los scripts a los que se apunta en el directorio **/etc/rc.d/init.d** son los caballos de trabajo; realizan el proceso real de iniciar y suspender los servicios. Cuando se ejecuta **/etc/rc.d/rc** a través de un directorio específico del nivel de ejecución, llama a cada script en un orden numérico. Primero ejecuta los scripts que empiezan con **K** y, a continuación, los que empiezan con **S**. Para los scripts que empiezan con **K**, se pasa un parámetro de **stop**. De modo semejante, para los scripts que empiezan con **S**, se pasa un parámetro de **start**.

Miremos dentro del directorio **/etc/rc.d/rc3.d** y veamos lo que está allí:

```
[root@serverA ~]# ls -l /etc/rc.d/rc3.d/
total 232
lrwxrwxrwx 1 root root 13 Jun 14 00:29 K01yum -> ../init.d/yum
lrwxrwxrwx 1 root root 24 Jun 14 00:29 K02NetworkManager -> ../init.d/Net...
lrwxrwxrwx 1 root root 34 Jun 14 00:29 K02NetworkManDisp.. -> ../init.d/...
lrwxrwxrwx 1 root root 19 Jun 14 00:28 K05saslauthd -> ../init.d/saslauthd
lrwxrwxrwx 1 root root 16 Jun 14 00:29 K10psacct -> ../init.d/psacct
lrwxrwxrwx 1 root root 14 Jun 14 00:29 S55sshd -> ../init.d/sshd
...<OUTPUT TRUNCATED>...
```

Al observar la salida muestra anterior, verá que “**K05saslauthd**” es uno de los muchos archivos que están en el directorio **/etc/rc.d/rc3.d** (línea 5). En consecuencia, cuando se ejecuta o llama el archivo **K05saslauthd**, el comando que en realidad más bien se está ejecutando es

```
#/etc/rc.d/init.d/saslauthd stop
```

Por el mismo motivo, si se llama **S55sshd**, el comando que sigue es el que en realidad se ejecuta:

```
#/etc/rc.d/init.d/sshd start
```

Escritura de su propio script rc

En el curso de mantener en ejecución un sistema Linux, en algún momento necesitará modificar el script de arranque y detención. Éstos son los dos caminos que puede tomar para hacer esto:

Si su cambio va a tener efecto sólo en el momento de la inicialización, y el cambio es pequeño, puede ser que quiera de manera escueta editar el script **/etc/rc.d/rc.local**. Este script se ejecuta justo al final del proceso de inicialización.

Por otra parte, si su adición es más elaborada y/o requiere que el proceso de paro se detenga explícitamente, debe agregar un script al directorio **/etc/rc.d/init.d**. Este script debe tomar los parámetros **start** y **stop**, y actuar en consecuencia.

Por supuesto, la primera opción, la edición del script **/etc/rc.d/rc.local**, es la más fácil de las dos. Para hacer adiciones a este script, sólo ábralo en su editor preferido y agregue los comandos que desea ejecutar al final. Esto es bueno para uno o dos cambios sencillos de línea.

Sin embargo, si en realidad necesita un script separado, necesitará tomar la segunda posibilidad. El proceso de escribir un script **rc** no es tan difícil como parece ser. Vayamos paso a paso a través de él con el uso de un ejemplo, para ver cómo funciona (a propósito, puede usar nuestro ejemplo como un script modelo, cambiándolo para agregar cualquier cosa que necesite).

Suponga que quiere arrancar un programa especial que haga aparecer en forma instantánea un mensaje cada hora que le recuerde que necesita tomar un descanso respecto al teclado (una

buenas ideas, ¡si no quiere padecer del síndrome del túnel carpiano!). El script para arrancar este programa incluirá lo que sigue:

- ▼ Una descripción de la finalidad del script (de manera que no la olvide pasado un año)
- Verificación de que el programa existe antes de intentar arrancarlo
- ▲ Aceptación de los parámetros **start** y **stop**, y desempeño de las acciones requeridas

NOTA Las líneas que empiezan con un signo de número (#) sólo son comentarios y no forman parte de las acciones del script, *excepto* para la primera línea.

Dados estos parámetros, empecemos a crear el script.

Creación del script **carpald.sh**

En primer lugar, crearemos el script que realizará la función real que deseamos. El script es muy simple, pero servirá para los fines que tenemos. En sus campos de comentarios, está incrustada una descripción de lo que el script hace.

1. Lance cualquier editor de textos que elija e introduzca el texto que sigue:

```
#!/bin/sh
#
#Descripción: Este sencillo script enviará un correo electrónico a cualquier
#dirección de correo especificada en la variable ADDR, cada hora, recordándole
#al usuario que se tome un descanso en relación con la computadora para evitar
#el síndrome del túnel carpiano. El script tiene una inteligencia tan pequeña que
#siempre enviará un correo tan pronto como el sistema esté listo y ejecutándose;
#;incluso cuando el usuario se quede dormido con rapidez! De modo que no olviden
#desactivarlo después del hecho.
#Autor: Wale Soyinka
#
ADDR=root@localhost
while true
    do
        sleep 1h
        echo "Get up and take a break NOW !!" | \
            mail -s "Carpal Tunnel Warning" $ADDR
    done
```

2. Guarde el texto del script en el archivo llamado **carpald.sh**.
 3. Enseguida, necesita hacer que el script sea ejecutable. Teclee
- ```
[root@serverA ~]# chmod 755 carpald.sh
```
4. Copie el script en el directorio en donde lo encontrarán los scripts de arranque, o muévalo a éste; es decir, al directorio **/usr/local/sbin/**. Teclee
- ```
[root@serverA ~]# mv carpald.sh /usr/local/sbin/
```

Creación del script de arranque

En este lugar, creará el script real de arranque que se ejecutará durante el inicio y paro del sistema. El archivo que creará aquí se llamará **carpald**. El archivo se activará con **chkconfig**. Esto quiere decir que, si deseamos, podemos usar la utilidad **chkconfig** para controlar los niveles de ejecución a los cuales se inicia y se detiene el programa. Ésta es una funcionalidad muy útil y que ahorra tiempo.

1. Lance cualquier editor de textos que elija e introduzca el texto que sigue:

```
#!/bin/sh
#Carpal      Start/Stop the Carpal Notice Daemon
#
#chkconfig: 35 99 01
# Descripción: Carpald es un programa que despierta cada hora y nos dice que
#               necesitamos tomar un descanso en relación con el teclado o
#               perderemos toda funcionalidad de nuestras muñecas y nunca seremos
#               capaces de teclear de nuevo mientras vivamos.
# Source function library.
. /etc/rc.d/init.d/functions

[ -f /usr/local/sbin/carpald.sh ] || exit 0

# See how we were called.
case "$1" in
    start)
        echo "Starting carpald: "
        /usr/local/sbin/carpald.sh &
        echo "done"
        touch /var/lock/subsys/carpald
        ;;
    stop)
        echo -n "Stopping carpald services: "
        echo "done"
        killall -q -9 carpald &
        rm -f /var/lock/subsys/carpald
        ;;
    status)
        status carpald
        ;;
    restart|reload)
        $0 stop
        $0 start
        ;;
*)
    echo "Usage: carpald start|stop|status|restart|reload"
    exit 1
esac
exit 0
```

Unos cuantos comentarios acerca del script de arranque anterior:

- ▼ Aun cuando la misma primera línea del script empieza con “#!/bin/sh”, se debe hacer notar que **/bin/sh** es un vínculo simbólico hacia **/bin/bash**. Éste no es el caso en otros sistemas UNIX.
- La línea “chkconfig: 35 99 01” en realidad es muy importante para la utilidad **chkconfig** que queremos usar. Los números “35” significan que **chkconfig** debe crear de modo predeterminado entradas de arranque y de detención para los programas en los niveles de ejecución 3 y 5; es decir, las entradas se crearán en los directorios **/etc/rc.d/rc3.d** y **/etc/rc.d/rc5.d**.
- ▲ Los campos “99” y “01” significan que **chkconfig** debe fijar la prioridad de arranque de nuestro programa en 99 y la detención en 01; es decir, el arranque tardío y el final anticipado.

2. Guarde el texto del script en un archivo llamado **carpald**.

3. A continuación, necesita hacer que el archivo sea ejecutable. Teclee

```
[root@serverA ~]# chmod 755 carpald
```

4. Copie o mueva el script en el directorio en donde se almacenan los scripts de arranque; por ejemplo, al directorio **/etc/rc.d/init.d/**. Teclee

```
[root@serverA ~]# mv carpald /etc/rc.d/init.d/
```

5. Ahora necesita decirle a **chkconfig** acerca de la existencia de este nuevo script de inicio/detención y lo que queremos hacer con él. Teclee

```
[root@serverA ~]# chkconfig --add carpald
```

Con esto se crearán en forma automática los vínculos simbólicos cuya lista damos a continuación para usted:

```
lrwxrwxrwx 1 root root 17 Dec 4 09:37 /etc/rc.d/rc0.d/K01carpald -> ../init.d/carpald
lrwxrwxrwx 1 root root 17 Dec 4 09:37 /etc/rc.d/rc1.d/K01carpald -> ../init.d/carpald
lrwxrwxrwx 1 root root 17 Dec 4 09:37 /etc/rc.d/rc2.d/K01carpald -> ../init.d/carpald
lrwxrwxrwx 1 root root 17 Dec 4 09:37 /etc/rc.d/rc3.d/S99carpald -> ../init.d/carpald
lrwxrwxrwx 1 root root 17 Dec 4 09:37 /etc/rc.d/rc4.d/K01carpald -> ../init.d/carpald
lrwxrwxrwx 1 root root 17 Dec 4 09:37 /etc/rc.d/rc5.d/S99carpald -> ../init.d/carpald
lrwxrwxrwx 1 root root 17 Dec 4 09:37 /etc/rc.d/rc6.d/K01carpald -> ../init.d/carpald
```

(Con anterioridad se explicó el significado y la importancia de los prefijos **K** [anular] y **S** [iniciar] en la lista anterior.)

Todo esto puede parecer más bien elaborado, pero las buenas noticias es que debido a que ha estructurado este script **rc**, nunca tendrá que hacerlo de nuevo. Lo que es más importante, el script se ejecutará en forma automática en el transcurso del arranque y paro, y será capaz de administrarse a sí mismo. ¡Lo en verdad superior es que bien valen la pena los beneficios a largo plazo de evitar el síndrome del túnel carpiano!

6. Use el comando **service** para averiguar el estado del programa **carpald.sh**. Teclee

```
[root@serverA ~]# service carpald status
carpald is stopped
```

7. En forma manual, inicie el programa **carpald** para estar seguro que en verdad arrancará en forma correcta al iniciar el sistema. Teclee

```
[root@serverA ~]# service carpald start
Starting carpald:
done
```

SUGERENCIA Si espera alrededor de una hora, debe ver un mensaje de correo proveniente del script **carpald.sh**. Puede usar el programa **mail** de la línea de comandos para teclear:

```
[root@serverA ~]# mail
Mail version 8.1 6/6/93. Type ? for help.
"/var/spool/mail/root": 1 message 1 new
>N 1 root@serverA.example Fri Dec 4 11:49 16/677 "Carpal Tunnel Warning"
&
```

Teclee **q** en el mensaje de ampersand (&) para abandonar el programa de correo.

8. A continuación, pare el programa. Teclee

```
[root@serverA ~]# service carpald stop
Stopping carpald services: done
```

9. Hemos terminado. Ahora es un buen momento para impedir que el programa **carpald** arranque con el sistema en el futuro. Teclee

```
[root@serverA ~]# chkconfig --del carpald
```

ACTIVACIÓN Y DESACTIVACIÓN DE SERVICIOS

A veces, puede encontrar que no necesita iniciar un servicio en particular en el momento de la inicialización. Esto se cumple en especial si está considerando Linux como un reemplazo para un archivo Windows NT y el servidor de impresión, y sólo necesita servicios muy específicos y nada más.

Como se describió en las secciones anteriores, puede hacer que no se inicie un servicio con sólo renombrar el vínculo simbólico en un directorio de nivel particular de ejecución; renómbrello para que empiece con una **K**, en lugar de una **S**. Una vez que se siente cómodo trabajando con la línea de comandos, hallará con rapidez que es muy fácil activar o desactivar un servicio.

También se pueden administrar los niveles de ejecución del servicio/programa en el arranque usando la utilidad **chkconfig**. Para ver todos los niveles de ejecución en los cuales está configurado el programa **carpald.sh** para arrancar, teclee

```
[root@serverA ~]# chkconfig --list carpald
carpald      0:off    1:off    2:off    3:on     4:off    5:on     6:off
```

Para hacer que el programa **carpald.sh** arranque en forma automática en el nivel 2, teclee

```
[root@serverA ~]# chkconfig --level 2 carpald on
```

Si revisa una vez más la lista de niveles de ejecución para el programa **carpald.sh**, verá que el campo para el nivel 2 se ha cambiado de 2:off (desactivado) a 2:on (activado). Teclee

```
[root@serverA ~]# chkconfig --list carpald
carpald      0:off   1:off   2:on    3:on    4:off   5:on    6:off
```

También se dispone de herramientas GUI que le ayudarán a administrar cuáles servicios arranquen en un nivel dado de ejecución. En Fedora y otros sistemas del tipo Red Hat (incluyendo RHEL), una de esas herramientas es la utilidad **system-config-services** (vea la figura 6-1). Para lanzar el programa, teclee

```
[root@serverA ~]# system-config-services
```

En un sistema SuSE Linux en ejecución, se puede lanzar el programa GUI equivalente (vea la figura 6-2) al teclear

```
serverA:~ yast2 runlevel
```

o bien,

```
serverA:~ yast runlevel
```

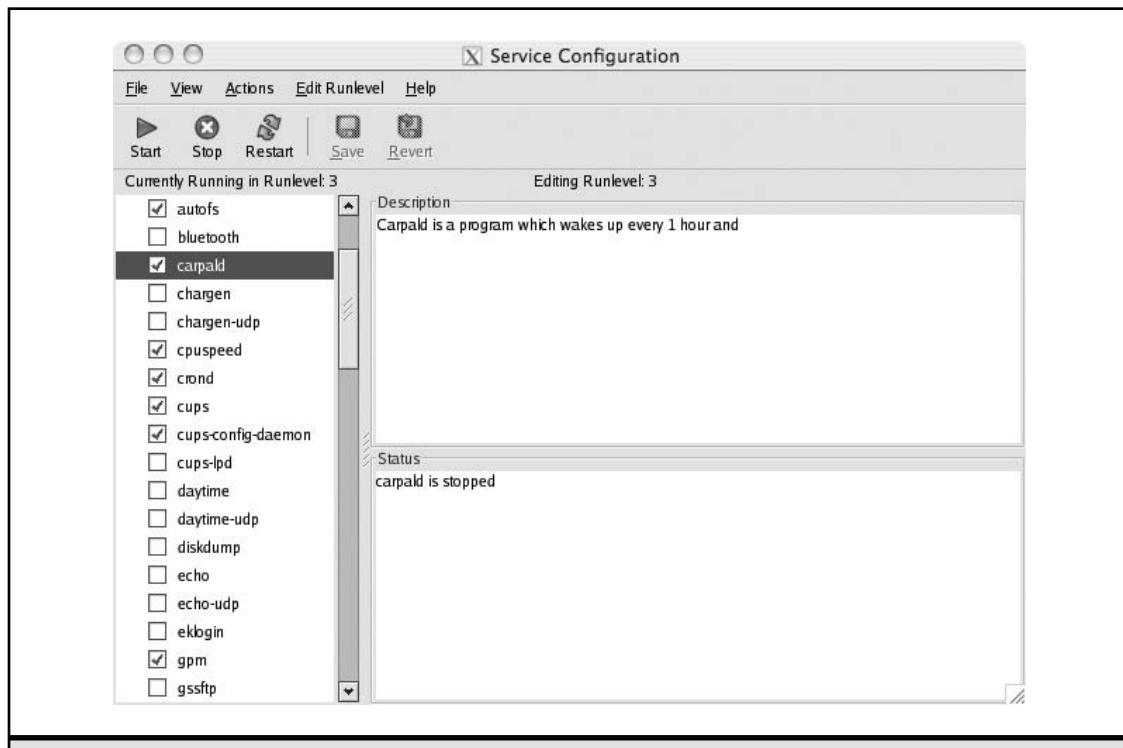


Figura 6-1. Herramienta GUI de configuración de servicios de Fedora Core

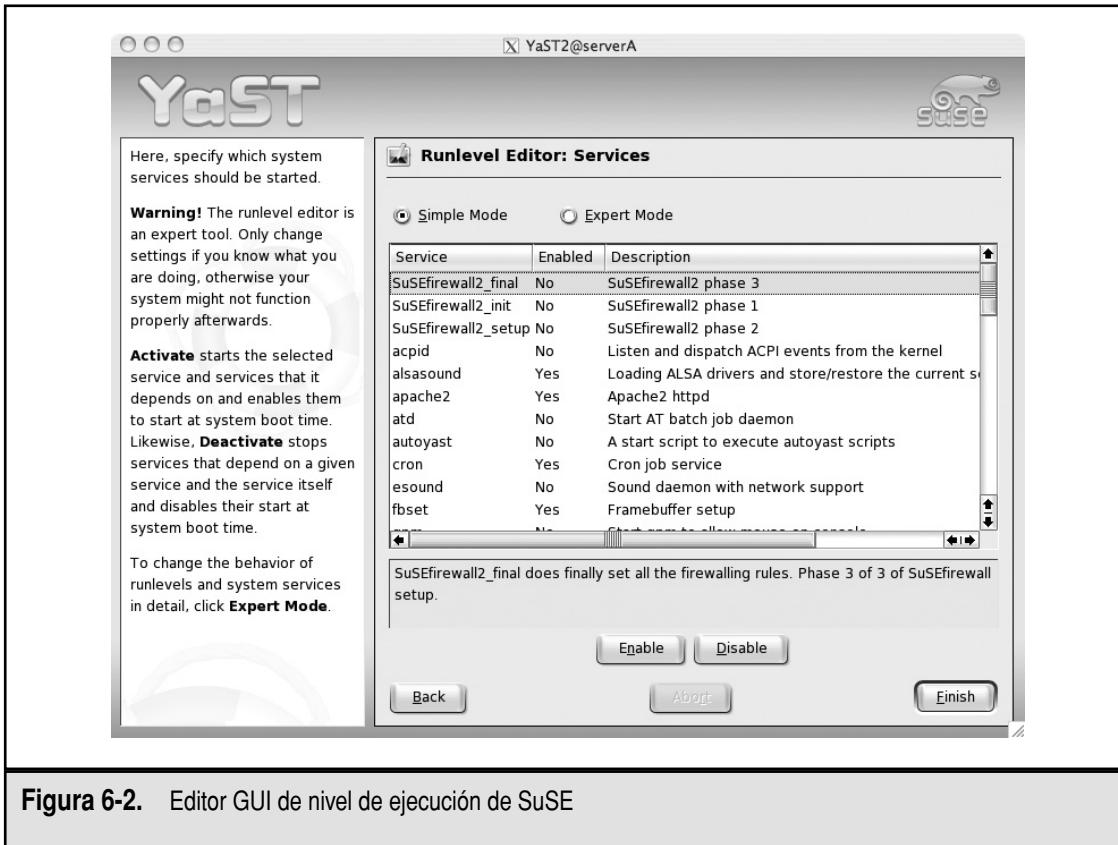


Figura 6-2. Editor GUI de nivel de ejecución de SuSE

Aunque la herramienta GUI es una manera bonita de llevar a cabo esta tarea, puede usted encontrarse en una situación en donde sólo no sea conveniente o no se disponga de ella.

Desactivación de un servicio

Para desactivar por completo un servicio, por lo menos debe conocer el nombre de ese servicio. Entonces puede usar la herramienta **chkconfig** para establecer de manera permanente que no se inicie en todos los niveles de ejecución.

Por ejemplo, para desactivar nuestro programa “salvavidas” **carpald.sh**, podría teclear

```
[root@serverA ~]# chkconfig carpald off
```

Si revisa una vez más la lista de niveles de ejecución del programa **carpald.sh**, verá que se ha desactivado para todos los niveles de ejecución. Teclee

```
[root@serverA ~]# chkconfig --list carpald
carpald      0:off    1:off    2:off    3:off    4:off    5:off    6:off
```

Para eliminar en forma permanente el programa **carpald.sh** de estar bajo el control de la utilidad **chkconfig**, usará la opción de borrar de esta utilidad. Teclee

```
[root@serverA ~]# chkconfig --del carpald
```

Hemos terminado con nuestro script muestra **carpald.sh** y para impedirle que nos inunde con notificaciones de correo electrónico en el futuro (en caso de que, de manera accidental, se activara de nuevo), podemos borrarlo del sistema para siempre. Teclee

```
[root@serverA ~]# rm -f /usr/local/sbin/carpald.sh
```

Y eso constituye el ABC de la manera de arrancar y parar servicios en forma automática en Linux. Ahora salgamos y tomemos un descanso.

SINGULARIDADES Y FINES DE LA INICIALIZACIÓN Y EL APAGADO

A la mayor parte de los administradores de Linux no les gusta apagar sus servidores Linux. Arruina su tiempo útil (recordará por lo visto en uno de los primeros capítulos que el “tiempo útil” es un motivo de orgullo para los administradores de sistemas Linux). Por consiguiente, cuando tiene que reiniciarse una caja Linux, suele ser por razones verdaderamente inevitables. Quizá ha sucedido algo malo o el núcleo se ha modernizado.

Por fortuna, Linux realiza un trabajo excelente de autorrecuperación, incluso durante los reinicios. Es raro tener que tratar con un sistema que no se inicializará en forma correcta, pero eso no quiere decir que nunca sucederá, y es eso a lo que esta sección se refiere.

¡fsck!

Estar seguro de que los datos en el disco duro de un sistema se encuentran en un estado coherente es una función muy importante. Esta función se controla de manera parcial por medio de un script de niveles de ejecución y otro archivo llamado **/etc/fstab**. La herramienta File System Check (**fsck**, Comprobación del sistema de archivos) se ejecuta en forma automática, como sea necesario, en cada inicialización, según se especifique por la presencia o ausencia de un archivo nombrado **./autofsck** y también según se especifique por medio del archivo **/etc/fstab**. La finalidad del programa **fsck** es semejante a la del Scandisk de Windows: revisar y reparar cualquier daño en el sistema de archivos, antes de continuar con el proceso de inicialización. En virtud de su naturaleza crítica, de manera tradicional, **fsck** se coloca muy al principio en la secuencia de arranque.

Si pudo hacer un paro limpio, el archivo **./autofsck** se borrará y **fsck** se ejecutará sin incidentes, como se especifique en el **/etc/fstab** (como se especifique en el sexto campo; vea la página de manual de fstab en man fstab). Sin embargo, si por alguna razón tuvo que realizar un paro difícil, (como el tener que presionar el botón de reinicialización), **fsck** necesitará ejecutarse recorriendo todos los discos locales cuya lista se encuentre en el archivo **/etc/fstab** y revisarlos (y no es raro que el administrador del sistema esté maldiciendo durante el curso del proceso).

Si, en realidad, **fsck** necesita ejecutarse, no entre en pánico. Es improbable que se le presente algún problema. Sin embargo, si de veras algo surge, **fsck** le pedirá información acerca del problema y le preguntará si quiere repararlo. En general, encontrará que responder “sí” es lo que debe hacerse.

La mayor parte de las distribuciones más recientes de Linux usa lo que se conoce como hacer un registro diario del sistema de archivos, y esto hace que sea fácil y más rápido recuperarse las

incoherencias de este sistema que podrían surgir provenientes de paros no limpios y otros pequeños errores del software. Ejemplos de sistemas de archivos con esta capacidad de registro diario son ext3, ReiserFS, jfs y xfs.

Por ejemplo, si está ejecutando el nuevo sistema de archivos ext3 o el ReiserFS, advertirá que la recuperación de reinicializaciones no limpias del sistema serán mucho más rápidas y fáciles. Lo único que hay que aceptar con la ejecución de un sistema de archivos con registro diario es la carga general que interviene en el mantenimiento del diario e, incluso, esto depende del método por el cual el sistema de archivos implementa su registro diario.

Inicialización en el modo de un solo usuario (“Recovery”)

Bajo Windows, el concepto de “Recovery Mode” se pidió prestado de una característica de hace largo tiempo de UNIX de inicializar en el modo de un solo usuario. Lo que esto significa para usted, bajo Linux, es que si algo se estropea en los scripts de arranque que afecte el proceso de inicialización de un anfitrión, es posible que inicialice en este modo, haga los arreglos y, a continuación, deje que el sistema se inicialice en un modo completo de usuarios múltiples (comportamiento normal).

Si está usando el cargador de inicialización GRUB, éstos son los pasos:

1. En primer lugar, necesita seleccionar la entrada de GRUB que desea inicializar, del menú del mismo GRUB, y enseguida oprimir la tecla **E**. A continuación, se le presentará un submenú con varias directivas (directivas del archivo **/boot/grub/menu.1st**).
2. Seleccione la etiqueta de entrada núcleo (kernel) y oprima **E** de nuevo. Ahora puede agregar la palabra clave **single** (o la letra **s**) al final de la línea. Presione **ENTER** para regresar al menú de inicialización de GRUB, después presione **B** para inicializar el núcleo en el modo de un solo usuario.
3. Cuando inicialice en el modo de un solo usuario, el núcleo de Linux se inicializará como normal, excepto cuando llega al punto en donde se inicia el programa **init**, sólo pasará por el nivel de ejecución 1 y, después, se detendrá (vea las secciones anteriores en este capítulo en relación con la descripción de todos los niveles de ejecución). Dependiendo de la configuración del sistema, se le pedirá la contraseña de la raíz o sólo se le dará un mensaje de shell. Si se le pide una contraseña, introduzca la contraseña del raíz y presione **ENTER**, y recibirá el mensaje de shell.
4. En este modo, encontrará que casi todos los servicios que de manera normal se inician no se están ejecutando. Esto incluye la configuración de la red. De modo que si necesita cambiar la dirección IP, la puerta de acceso, la netmask o cualquier archivo de configuración relacionado con la red, usted puede hacerlo. Éste también es un buen momento para ejecutar **fsck** en forma manual en cualquier partición que pudiera no haber sido revisada y recuperada en forma automática (el programa **fsck** le dirá cuáles particiones se están comportando mal, si las hay).

SUGERENCIA En el modo de un solo usuario de muchas distribuciones de Linux, sólo se montará en forma automática la partición raíz para usted. Si necesita tener acceso a cualesquier otras particiones, necesitará montarlas por sí mismo usando el comando **mount**. Puede ver todas las particiones que puede montar en el archivo **/etc/fstab**.

5. Una vez que haya hecho cualesquiera cambios que necesite hacer, sencillamente oprima CTRL-D. Esto lo hará salir del modo de un solo usuario y continuar con el proceso de inicialización, o bien, puede emitir tan sólo el comando de reiniciar para que se reinicie el sistema.

RESUMEN

En este capítulo, se analizaron los diversos aspectos relacionados con el arranque y el paro de un sistema Linux típico. Empezamos nuestro examen con el todopoderoso cargador de inicialización. Vimos GRUB en particular, como un cargador/administrador de inicialización muestra porque es el cargador elegido entre las distribuciones más populares de Linux. A continuación, examinamos cómo es típico que se inicien y detengán las cosas (o servicios) en Linux y de qué manera decide éste lo que se inicia y se detiene, y en cuál nivel de ejecución se supone que hace esto. Incluso escribimos un pequeño programa shell, como una demostración, que nos ayuda a evitar el síndrome del túnel carpiano. Después, seguimos adelante y configuramos el sistema para arrancar automáticamente el programa en niveles específicos de ejecución.

CAPÍTULO 7



Sistemas
de archivos

Los sistemas de archivos son los mecanismos mediante los cuales se organizan los datos en un medio de almacenamiento. Proporcionan todas las capas de separación arriba de los sectores y cilindros de los discos. En este capítulo, discutiremos la composición y administración de estas capas de separación soportadas por Linux. Se le pondrá una atención particular al sistema predeterminado de archivos de Linux, ext2/ext3.

También se cubrirán los muchos aspectos de la administración de discos. Esto incluye la creación de particiones y volúmenes, el establecimiento de sistemas de archivos, la automatización del proceso por el cual se montan en el momento de la inicialización y tratar con ellos después de que se cae un sistema. También se tocarán los conceptos de Logical Volume Management (LVM, Administración de volúmenes lógicos).

NOTA Antes de empezar el estudio de este capítulo, ya debe estar familiarizado con los archivos, directorios, permisos y propiedad en el entorno de Linux. Si todavía no ha leído el capítulo 5, lo mejor es que lo lea antes de continuar.

LA COMPOSICIÓN DE LOS SISTEMAS DE ARCHIVOS

Empecemos por pasar por la estructura de los sistemas de archivos bajo Linux. Ayudará a aclarar su comprensión del concepto y le permitirá ver con más facilidad cómo obtener ventaja de la estructura.

Nodos i

El bloque de construcción más fundamental de muchos sistemas de archivos de UNIX (incluyendo ext2 de Linux) es el *nodo i*. Un nodo i es una estructura de control que apunta hacia otros nodos i o hacia los bloques de datos.

La información de control en el nodo i incluye el propietario del archivo, los permisos, el tamaño, la hora del último acceso, la hora de creación, la ID del grupo, etcétera (para los verdaderamente curiosos, la estructura completa de los datos del núcleo se encuentra en `/usr/src/linux/include/linux/ext2_fs.h`; suponiendo, por supuesto, que tiene instalado el árbol fuente en el directorio `/usr/src`). Lo único que *no* conserva un nodo i es el nombre del archivo.

Como se mencionó en el capítulo 5, los propios directorios son ejemplos especiales de archivos. Esto significa que cada directorio posee un nodo i, y este nodo i apunta hacia los bloques de datos que contienen información (nombres de archivos y nodos i) referente a los archivos que se encuentran en el directorio. En la figura 7-1, se ilustra la organización de los nodos i y los bloques de datos en el sistema de archivos ext2.

Como puede ver en la figura 7-1, los nodos i se usan para proporcionar *caminos indirectos* de modo que se puede apuntar hacia más bloques de datos; lo cual es la razón por la que cada nodo i no contiene el nombre del archivo (sólo un nodo i funciona como representativo para el archivo completo; como consecuencia, sería un desperdicio de espacio si cada nodo i contuviera la información del nombre de archivo). Tome por ejemplo un disco de 6GB que contiene 1 079 304 nodos i. Si cada uno de estos nodos consumiera 256 bytes para almacenar el nombre del archivo, se desperdiciarían un total de alrededor de 33MB en el almacenamiento de los nombres de archivos, ¡incluso si no se estuvieran usando!

A su vez, si es necesario, cada bloque indirecto puede apuntar hacia otros bloques indirectos. Con hasta tres capas de caminos indirectos, es posible almacenar archivos con tamaños muy grandes en un sistema de archivos de Linux.

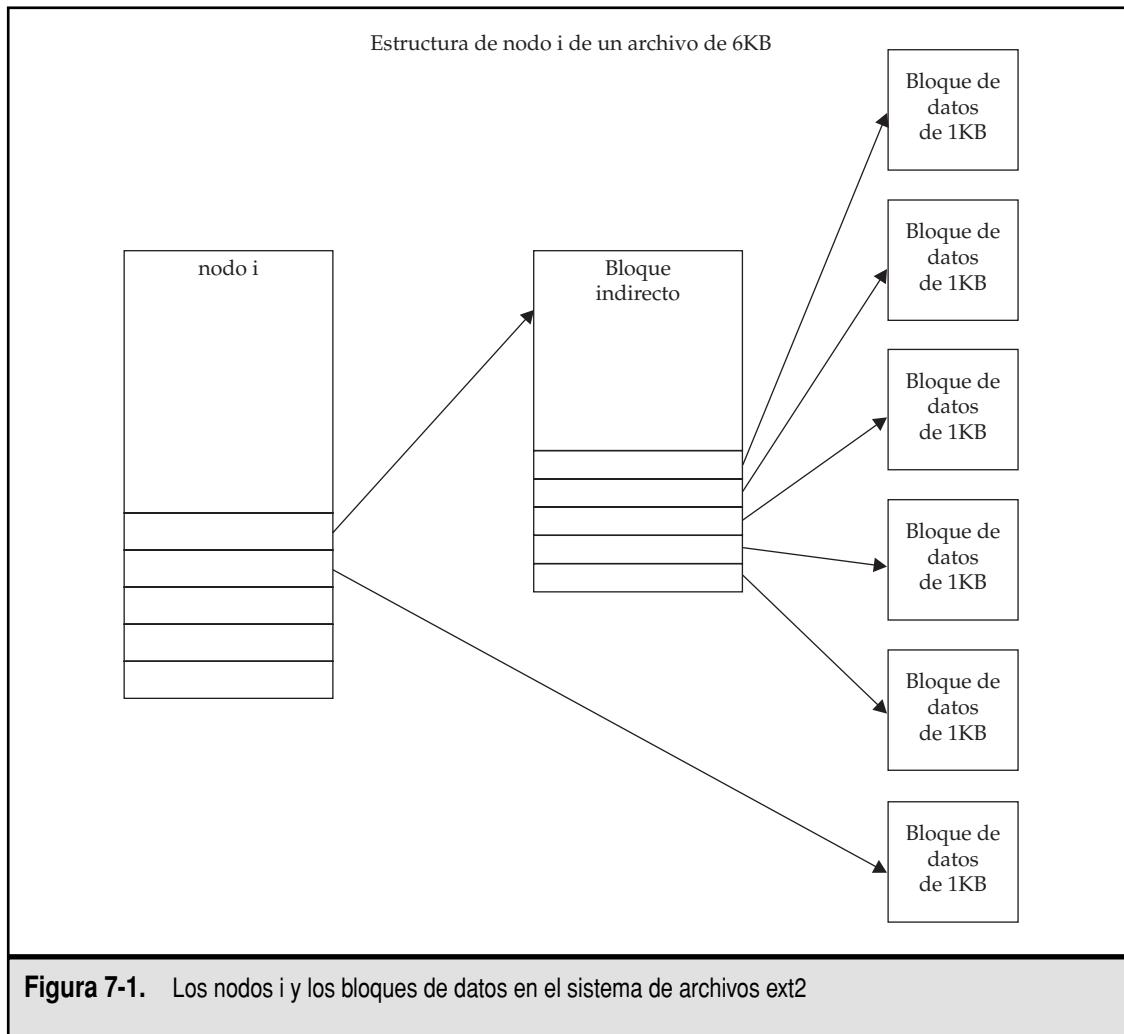


Figura 7-1. Los nodos i y los bloques de datos en el sistema de archivos ext2

Superbloques

La precisa primera pieza de información que se lee de un disco es su *superbloque*. Esta pequeña estructura de datos revela varias piezas clave de información, incluyendo la configuración geométrica del disco, la cantidad de espacio disponible y, lo que es más importante, la ubicación del primer nodo i. Sin un superbloque, un sistema de archivos en un disco es inútil.

Algo tan importante como el superbloque no se deja al azar. Múltiples copias de esta estructura de datos se dispersan por todo el disco para suministrar respaldo en caso de que se dañe el primero. Bajo el sistema de archivos ext2 de Linux, se coloca un superbloque después de cada *grupo* de bloques, el cual contiene los nodos i y los datos. Un grupo consta de 8 192 bloques; por tanto, el primer superbloque redundante está en el 8 193, el segundo en el 16 385, y así sucesivamente. Los diseñadores de la mayor parte de los sistemas de archivos de Linux incluyeron de manera inteligente esta redundancia del superbloque en el diseño de esos sistemas.

ext3 y ReiserFS

Ext3 y ReiserFS son dos sistemas populares de archivos de Linux, usados en la mayor parte de las distribuciones principales de éste. El sistema ext3 de archivos es una extensión mejorada del sistema ext2. En la época en que se está escribiendo este libro, el sistema ext2 de archivos tiene alrededor de 14 años. Esto significa dos cosas para nosotros, como administradores de sistemas. La primera y principal, ext2 es roca sólida. Es un subsistema bien probado de Linux y ha tenido el tiempo para ser muy bien optimizado. La segunda, otros sistemas de archivos que se consideraban experimentales cuando se creó ext2 han madurado y han llegado al punto de encontrarse a disponibilidad de Linux.

Los dos sistemas de archivos que son reemplazos populares para el ext2 son el ext3 y ReiserFS. Los dos ofrecen mejoras significativas en el rendimiento y la estabilidad, pero la componente más importante de los dos es que se han movido hacia un nuevo método de captar los datos en el disco. Este nuevo método se llama *registro diario (journaling)*. En los sistemas tradicionales de archivos (como el ext2) se debe buscar por toda la estructura del directorio, encontrar el lugar correcto en el disco para disponer los datos y, a continuación, disponer de ellos (en Linux también se puede usar la caché para el proceso completo, incluyendo las actualizaciones de los directorios, haciendo de este modo que al usuario le parezca más rápido el proceso). Casi todas las nuevas versiones de las distribuciones de Linux ahora hacen uso de modo predeterminado de uno de los sistemas de archivos de registro diario o del otro. Por ejemplo, en Fedora Core y los diversos productos de RHEL se usa ext3 de modo predeterminado y, en SuSE se usa ReiserFS.

El problema de no contar con un sistema de archivos de registro diario es que, en el caso de una caída inesperada del sistema, el programa verificador del sistema de archivos o verificador de la coherencia de ese sistema (**fsck**) tiene que hacer el seguimiento de todos los archivos que están en el disco para asegurarse que no contienen algunas referencias colgantes (por ejemplo, nodos i que apuntan hacia otros nodos i o bloques de datos inválidos). Conforme los discos se expanden en tamaño y se contraen en precio, la disponibilidad de estos discos de gran capacidad significa que más de nosotros tendremos que tratar con las repercusiones de tener que aplicar el **fsck** a un disco grande. Y como cualquiera que ha tenido que hacer esto con anterioridad le puede decir, no es divertido. El proceso puede tardar un tiempo largo para completarse, y eso significa tiempo muerto para sus usuarios.

Los sistemas de archivo con registro diario trabajan creando en primer lugar una entrada de clasificaciones en un registro (o diario) de cambios que están a punto de hacerse, antes de en realidad consignar los cambios al disco. Una vez que la transacción se ha consignado al disco, entonces el sistema de archivos sigue adelante para modificar los datos reales o metadatos. Esto conduce a una situación de todo o nada; es decir, se realizan todos o ninguno de los cambios en el sistema de archivos.

Uno de los beneficios de usar un sistema de archivos del tipo de registro diario es la mayor seguridad de que se conservará la integridad de los datos, y en las situaciones inevitables, en donde surgen los problemas, la velocidad, la facilidad de la recuperación y la posibilidad de tener éxito se aumentan de manera importante. Una de esas situaciones inevitables podría ser en el caso de una caída del sistema. En este caso, puede ser que no necesite ejecutar **fsck**. ¡Piense en cuánto más rápido podría recuperar un sistema si no tuviera que ejecutar de 200GB! (¿No ha tenido que ejecutar antes **fsck** en un disco grande? Piense en cuánto tarda en ejecutarse Scandisk bajo Windows en discos grandes.) Otros beneficios de usar los sistemas de archivos del tipo de registro diario son que se simplifican las reiniciaciones del sistema, se reduce la fragmentación del disco y se pueden acelerar las operaciones de entrada/salida (I/O) (esto depende del método de registro diario que se use).

Si quiere aprender más acerca del sistema de archivos ext2, recomendamos que lea la edición más reciente del libro titulado *Linux Kernel Internals*, editado por Michael Beck (Addison-Wesley, 1998). Aunque el libro es muy anticuado en muchos aspectos, en términos del núcleo (fue escrito para la serie 2.0), las partes relativas al sistema de archivos ext2 todavía se cumplen, ya que ext2 es la base del sistema ext3.

¿Cuál sistema de archivos usar?

El lector por ahora podría estar preguntando: ¿cuál sistema de archivos debo usar? En el momento en que se está escribiendo este libro, la tendencia actual es la de desplazarse hacia cualquier sistema de archivos con capacidades de registro diario. Como con todas las cosas de Linux, usted elige. Su mejor apuesta es probar muchos sistemas de archivos y ver cómo se comportan con la aplicación para la que usted está usando el sistema. Sólo tenga presente que el registro diario tiene su propia carga general.

ADMINISTRACIÓN DE LOS SISTEMAS DE ARCHIVOS

El proceso de administrar los sistemas de archivos es trivial; es decir, la administración se vuelve trivial *después* que ha memorizado todos los aspectos de sus servidores en red, discos, respaldos y requisitos de tamaño, con la condición de que nunca tendrá nuevamente que hacer cambios. En otras palabras, la administración de los sistemas de archivos no es trivial en lo absoluto.

Una vez que los sistemas de archivos se han creado, desplegado y agregado al ciclo de respaldo, tienden, en su mayor parte, a cuidar de sí mismos. Lo que los hace difíciles de administrar son los aspectos administrativos: como los usuarios que se niegan a limpiar sus discos y las engorrosas políticas de administración que establecen quién puede compartir cuál disco y con cuáles condiciones, dependiendo, por supuesto, de la cuenta con la cual se compró el almacenamiento/disco y... (suena de manera espantosa, como una tira cómica de *Dilbert*, pero existe una gran cantidad de verdad detrás de lo antes dicho).

Por desgracia, no se dispone de una solución que se encuentre en un recetario de cocina para tratar con las políticas de oficina, de modo que, en esta sección, nos apegaremos a los aspectos técnicos relacionados con la administración de los sistemas de archivos; es decir, el proceso de montar y desmontar particiones, tratar con el archivo `/etc/fstab` y realizar la recuperación del sistema de archivos con la herramienta `fsck`.

Montaje y desmontaje de los discos locales

Los puntos fuertes de Linux incluyen su flexibilidad y la manera en la que se presta a la administración sin costuras de las ubicaciones de los archivos. Las particiones necesitan montarse de modo que se pueda tener acceso a su contenido (en realidad, lo que se monta es el sistema de archivos en una partición o volumen). Los sistemas de archivos se montan de modo que se vean sólo como otro subdirectorio en el sistema. Esto ayuda a fomentar la ilusión de una estructura grande de directorios, aun cuando puede haber varios sistemas diferentes de archivos en uso. Esta característica resulta útil en especial para el administrador, quien puede reubicar los datos almacenados en una partición física hacia una nueva ubicación (quizá en una partición diferente) bajo el árbol de directorios, sin que ninguno de los usuarios del sistema sea el más sabio.

El proceso de administración del sistema de archivos empieza con el directorio raíz. Esta partición también se conoce como barra diagonal y, del mismo modo, se simboliza por medio de

una barra diagonal (/). La partición que contiene la estructura del núcleo y el directorio central se monta en el momento de la inicialización. Es posible y usual que la partición física que aloja el núcleo de Linux esté en un sistema separado de archivos, como **/boot**. También es posible que el sistema de archivos raíz (“/”) aloje tanto al núcleo y otras utilidades requeridas como a los archivos de configuración para llevar al sistema hasta un modo de un solo usuario.

A medida que se ejecutan los scripts de inicialización, se montan sistemas adicionales de archivos, agregándolos a la estructura del sistema de archivos raíz. El proceso de montaje sobrepone un solo subdirectorio con el árbol de directorios que está tratando de montar. Por ejemplo, digamos que /dev/hda2 es la partición raíz. Tiene el directorio **/usr**, el cual no contiene archivos. La partición /dev/hda3 contiene todos los archivos que usted quiere en **/usr**, de modo que monte /dev/hda3 en el directorio **/usr**. Ahora los usuarios pueden sencillamente cambiar los directorios a **/usr** para ver todos los archivos de esa partición. El usuario no necesita saber que **/usr** en realidad es una partición separada.

NOTA En éste y en otros capítulos podríamos decir de manera inadvertida que se está montando una partición en tal y tal directorio. Por favor, note que, en realidad, es el sistema de archivos en la partición el que se está montando. En beneficio de la sencillez y manteniéndose en el modo de hablar cotidiano, podríamos intercambiar estos dos significados.

Tenga presente que cuando se monta un directorio nuevo, el proceso **mount** esconde todo el contenido del directorio montado con anterioridad. De manera que en nuestro ejemplo del **/usr**, si la partición raíz en realidad tuvo archivos en **/usr**, antes del montaje de /dev/hda3, esos archivos de **/usr** ya no serían visibles (por supuesto, no se han borrado; una vez que se desmontara /dev/hda3, los archivos de **/usr** estarían visibles de nueva cuenta).

Uso del comando **mount**

Como muchas herramientas de la línea de comandos, el comando **mount** tiene una abundancia de opciones, la mayor parte de las cuales usted no estará usando en el trabajo diario. Puede conocer los detalles completos relativos a estas opciones en la página man de **mount**. En esta sección, examinaremos los usos más comunes del comando.

La estructura del comando **mount** es la siguiente:

```
mount [options] device directory
```

en donde **options** pueden ser cualesquiera de las que se muestran en la tabla 7-1.

En la tabla 7-2, se muestran las opciones de las que se dispone para usarse con la bandera **mount -o**.

Si se emite el comando **mount** sin opciones, presentará una lista de todos los sistemas de archivos montados hasta el momento. Por ejemplo, teclee

```
[root@serverA ~]# mount
/dev/mapper/VolGroup00-LogVol00 on / type ext3 (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
... (OUTPUT TRUNCATED) ...
/dev/mapper/VolGroup00-LogVol03 on /tmp type ext3 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
```

Opción para <code>mount</code>	Descripción
<code>-a</code>	Monta todos los sistemas de archivos cuya lista se da en <code>/etc/fstab</code> (este archivo se examina más adelante en esta sección).
<code>-t <i>fstype</i></code>	Especifica el tipo de sistema de archivos que se está montando. En Linux se pueden montar sistemas de archivos que sean diferentes al ext2 estándar, con la mayor frecuencia FAT, VFAT y FAT32. El comando <code>mount</code> suele sentir esta información por sí mismo.
<code>-o <i>options</i></code>	Especifica las opciones que se están aplicando a este proceso <code>mount</code> . Por lo común, éstas son opciones específicas para el tipo de sistema de archivos (es posible que las opciones para montar sistemas de archivos a la red no se apliquen para montar sistemas locales de archivos).

Tabla 7-1. Opciones de las que se dispone para el comando `mount`

Opción para el parámetro <code>mount -o</code> (para particiones locales)	Descripción
<code>ro</code>	Monta la partición como sólo lectura.
<code>rw</code>	Monta la partición como lectura/escritura (predeterminado).
<code>exec</code>	Permite la ejecución de binarios (predeterminado).
<code>noatime</code>	Desactiva la actualización de la hora de acceso en los nodos i. Para las particiones en donde no importa la hora de acceso, desactivar esto mejora el rendimiento.
<code>noauto</code>	Desactiva el montaje automático de esta partición cuando se especifica la opción <code>-a</code> (sólo se aplica al archivo <code>/etc/fstab</code>).
<code>nosuid</code>	No permite la aplicación de los bits del programa SetUID a la partición montada.
<code>sb=n</code>	Le dice a <code>mount</code> que use el bloque <code>n</code> como el superbloque. Esto es útil cuando podría dañarse el sistema de archivos.

Tabla 7-2. Opciones de las que se dispone para usarse con el parámetro `mount -o`

El siguiente comando **mount** monta la partición /dev/hda3 sobre el directorio /bogus-directory con privilegios de sólo lectura:

```
[root@serverA ~]# mount -o ro /dev/hda3 /bogus-directory
```

Desmontaje de los sistemas de archivos

Para desmontar un sistema de archivos, use el comando **umount** (note que el comando *no* es **unmount**). Enseguida se da el formato del comando:

```
umount [-f] directory
```

en donde **directory** es el directorio que se va a desmontar. Por ejemplo,

```
[root@serverA ~]# umount /bogus-directory
```

desmonta la partición montada en el directorio /bogus-directory.

Cuando el sistema de archivos está en uso Existe una trampa para **umount**: si el sistema de archivos está en uso (es decir, cuando alguien está en ese momento teniendo acceso al contenido de ese sistema de archivos o tiene un archivo abierto en éste), no podrá desmontar ese sistema. Para darle la vuelta a esto, tiene las posibilidades siguientes:

- ▼ Puede usar el programa **lsof** o **fuser** para determinar cuáles procesos están manteniendo abiertos los archivos y, a continuación, anularlos o pedir a los propietarios del proceso que suspendan lo que están haciendo (lea acerca del parámetro **kill** en **fuser**, en la página man de este último). Si decide anular los procesos, asegúrese que entiende las repercusiones de hacerlo (lea: que no lo despidan por hacer esto).
- Puede usar la opción **-f** con **umount** para forzar a que el proceso se desmonte. Esto es en especial útil para los sistemas de archivos del tipo NFS de los que ya no se dispone.
- Use el desmontaje Lazy. Esto se especifica con la opción **-l**. Esta opción casi siempre funciona incluso cuando las otras fallan. Separa de inmediato el sistema de archivos de la jerarquía de sistemas de archivos y limpia todas las referencias hacia ese sistema tan pronto como deja de estar ocupado.
- ▲ La alternativa más segura y más apropiada es bajar el sistema hasta el modo de un solo usuario y, después, desmontarlo. Por supuesto, en realidad no siempre se puede dar ese lujo.

El archivo /etc/fstab

Como se mencionó al principio, /etc/fstab es un archivo de configuración que **mount** puede usar. Este archivo contiene una lista de todas las particiones conocidas del sistema. Durante el proceso de inicialización, esta lista se lee y los ítems que se encuentran en ella se montan en forma automática, con las opciones especificadas al respecto.

He aquí el formato de las entradas en el archivo /etc/fstab:

/dev/device	/dir/to/mount	fstype	parameters	fs_freq	fs_passno
-------------	---------------	--------	------------	---------	-----------

El siguiente es un archivo **/etc/fstab** muestra:

```

1. /dev/VolGroup00/LogVol00 /
2. LABEL=/boot           /boot      ext3    defaults        1  1
3. none                 /dev/pts   devpts  gid=5,mode=620  0  0
4. none                 /dev/shm   tmpfs   defaults        0  0
5. /dev/VolGroup00/LogVol02 /home     ext3    defaults        1  2
6. none                 /proc      proc    defaults        0  0
7. none                 /sys       sysfs  defaults        0  0
8. /dev/VolGroup00/LogVol03 /tmp      ext3    defaults        1  2
9. /dev/VolGroup00/LogVol01 swap     swap    defaults        0  0
10. /dev/hdc   /media/cdrom   auto
    pamconsole,fscontext=system_u:object_r:removable_t,ro,exec,noauto,managed 0  0
11. /dev/fd0   /media/floppy  auto
    pamconsole,fscontext=system_u:object_r:removable_t,exec,noauto,managed 0  0

```

Echemos una mirada a algunas de las entradas en el archivo **/etc/fstab** que todavía no se han discutido. Por favor, note que se han agregado números de línea a la salida anterior para ayudar a su legibilidad.

Línea 1 La primera entrada en nuestro archivo **/etc/fstab** muestra es la correspondiente al volumen raíz. En la primera columna se muestra el dispositivo que aloja el sistema de archivos; es decir, el volumen lógico **/dev/VolGroup00/LogVol00** (más adelante se hablará más sobre volúmenes). En la segunda se muestra el punto de montaje; es decir, el directorio **“/”**. En la tercera se tiene el tipo de sistema de archivos; es decir, **ext3**, en este caso. En la cuarta se dan las opciones con las cuales debe montarse el sistema de archivos; en este caso, sólo se requieren las opciones predeterminadas. El quinto campo se usa para dar cabida a la utilidad **dump** (una sencilla herramienta de respaldo que se discute en el capítulo 28), para determinar cuáles sistemas de archivos necesitan respaldarse. Y el sexto campo y final se usa para el programa **fsck**, a fin de determinar si se necesita verificar el sistema de archivos y también para determinar el orden en el cual se hacen las verificaciones.

Línea 2 La entrada que sigue en nuestro archivo ejemplo es el punto de montaje **/boot**. En el primer campo de esta entrada se muestra el dispositivo; en este caso, apunta hacia cualquier dispositivo con la etiqueta **/boot**. Los otros campos tienen un significado que básicamente es lo mismo que lo correspondiente al campo para el punto de montaje raíz que se discutió con anterioridad.

Note que en el caso del punto de montaje **/boot**, podría advertir que el campo para el dispositivo se mira diferente a la convención usual **/dev/<path-to-device>** (**/dev/<trayectoria al dispositivo>**). El uso de etiquetas ayuda a esconder el dispositivo (partición) real del que el sistema de archivos se está montando. El dispositivo se ha reemplazado con un símbolo que se mira como lo siguiente: **LABEL=/boot**. En el curso de la instalación inicial, el programa que realiza la partición del instalador fija la etiqueta en esa partición. Después de la inicialización, el sistema recorre las tablas de particiones y busca estas etiquetas. Esto resulta especialmente útil cuando se están usando discos SCSI. Por lo común, el SCSI tiene una SCSI ID fijada. Con el uso de etiquetas, puede mover el disco por ahí y cambiar la SCSI ID, y el sistema todavía sabrá cómo montar el sistema de archivos, aun cuando el dispositivo podría haber cambiado de, por ejemplo, **/dev/sda10** a **/dev/sdb10** (vea la sección “Convenciones acerca del nombramiento de discos y particiones”, más adelante).

Línea 4 A continuación viene el sistema de archivos tmpfs, también conocido como sistema de archivos de memoria virtual (VM, *virtual memory*). Éste usa tanto la RAM del sistema como el área de intercambio. No es un dispositivo típico de bloques, porque no existe sobre la parte superior de un dispositivo subyacente de bloques; está asentado directamente sobre la parte superior de la VM. Se usa para solicitar páginas del subsistema VM para almacenar archivos. El primer campo –**none**– significa que esta entrada trata con una VM y, como tal, no está asociada con archivo de dispositivo normal UNIX/Linux. En la segunda entrada, se muestra el punto de montaje, /dev/shm. En el tercer campo se da el tipo de sistema de archivos; es decir, tmpfs. En el cuarto campo se hace ver que este sistema de archivos debe montarse con las opciones predeterminadas. El quinto y sexto campos tienen el mismo significado que aquellos discutidos para las entradas anteriores. Note en especial que, en este caso, los valores son cero, lo cual tiene un sentido perfecto porque no hay razón para ejecutar un programa no inteligente en un sistema de archivos temporales en la inicialización y tampoco hay razón para ejecutar **fsck** en él, dado que no contiene un sistema de archivos del tipo ext2/3.

Línea 6 La siguiente entrada notable es para el sistema de archivos del tipo proc. En este sistema de archivos, se mantiene dinámicamente la información referente a los procesos del sistema (de allí la abreviatura proc). El **none** en el primer campo de la entrada proc, en el archivo /etc/fstab, tiene la misma implicación que la de la entrada del sistema de archivos tmpfs. El sistema de archivos proc es un sistema especial que proporciona una interfaz para los parámetros del núcleo a través de la que se mira como cualquier otro sistema de archivos; es decir, proporciona una visión casi legible para las personas del núcleo. Aunque parece que existe en el disco, en realidad no lo está; todos los archivos representan algo que está en el núcleo. El más notable es /dev/kcore, el cual es la memoria del sistema resumida como archivo. A menudo, la gente para la que el sistema de archivos proc es nuevo considera en forma equivocada que éste es un archivo grande e innecesario y lo eliminan de manera accidental, lo cual causará un mal funcionamiento del sistema de muchas maneras gloriosas. A menos que esté seguro que sabe lo que está haciendo, resulta una apuesta segura solamente dejar todos los archivos del directorio /proc (en el capítulo 10, aparecen más detalles del /proc).

Línea 7 Sigue la entrada para el sistema de archivos sysfs. Éste es nuevo y necesario en los núcleos Linux 2.6. Una vez más, se trata de uno temporal y especial, precisamente como los sistemas de archivos tmpfs y proc. Sirve como un depósito interno de memoria para la información del estado del sistema y de los dispositivos. Suministra una visión estructurada del árbol de dispositivos de un sistema. También es semejante a ver los dispositivos en el Device Manager (Administrador de dispositivos) de Windows como una serie de archivos y directorios, en lugar de a través de la vista del panel de control.

Línea 8 La entrada siguiente es para el punto de montaje /tmp. Esto se refiere a una entidad o un dispositivo físicos reales en el sistema, precisamente como el punto de montaje raíz (“/”) y el /boot.

Línea 9 Ésta es la entrada para la partición de intercambio (swap) del sistema. Es donde reside la memoria virtual. En Linux, la memoria virtual se puede conservar en una partición separada de la partición raíz (se debe hacer notar que en Linux, para los fines del intercambio, también se puede usar un archivo común). Mantener el espacio de intercambio en una partición separada ayuda a mejorar el rendimiento, ya que la partición de intercambio puede obedecer reglas de manera diferente que un sistema normal de archivos. Asimismo, puesto que la partición no necesita ser respal-

dada o verificada con **fsck** en el momento de la inicialización, los dos últimos parámetros en ella se hacen iguales a cero (note que una partición de intercambio se puede mantener también en un archivo de disco normal. Para obtener información adicional, vea la página man en **mkswap**).

Línea 10 La última entrada en el archivo **fstab** que quizá vale la pena mencionar es la de los medios removibles. En este ejemplo, el campo de dispositivo señala el archivo de dispositivo que representa el dispositivo cdrom. En este caso, la unidad de CD-ROM es el master del controlador IDE secundario (/dev/hdc). El punto de montaje es /media/cdrom y, por tanto, cuando se inserta un CD-ROM y se monta en el sistema, se puede tener acceso al contenido del CD desde el directorio /media/cdrom. El **auto** en el tercer campo significa que el sistema intentará de manera automática sondear/detectar el tipo correcto del sistema de archivos para el dispositivo. Para los CD-ROM, éste suele ser el sistema de archivos iso9660. En el cuarto campo, se da una lista de las opciones de montaje. Estas opciones particulares son pertinentes en un sistema con un núcleo activado por SELinux.

NOTA Al montar las particiones con el archivo /etc/fstab configurado, puede ejecutar el comando **mount** con sólo un parámetro: el directorio en el que desea montarlas. El comando **mount** revisa /etc/fstab en busca de ese directorio; si lo encuentra, **mount** usará todos los parámetros que ya hayan sido establecidos allí. Por ejemplo, he aquí el comando corto para montar el CD-ROM, dado el archivo /etc/fstab mostrado al principio:

```
[root@serverA ~]# mount /media/cdrom/
```

Uso de fsck

La herramienta **fsck** (abreviatura de File System Check, Verificación de los sistemas de archivos) se usa para diagnosticar y reparar los sistemas de archivos que pueden haber resultado dañados en el curso de las operaciones diarias. Esas reparaciones suelen ser necesarias después de que un sistema se cae, en tal forma que no dio posibilidad de vaciar por completo sus memorias intermedias (*buffers*) hacia el disco (aun cuando el nombre de esta herramienta guarda una sorprendente semejanza a una de las expresiones que a menudo se pronuncian después de que un sistema se cae, que esta herramienta forme parte del proceso de recuperación es *estrictamente* una coincidencia).

Por lo común, el sistema ejecuta la herramienta **fsck** en forma automática en el proceso de inicialización, ya que se estima necesario (de manera muy semejante a la que Windows ejecuta Scandisk). Si detecta un sistema de archivos que no se desmontó con limpieza, ejecuta la utilidad. Linux realiza un esfuerzo impresionante para reparar en forma automática cualesquiera problemas por los que pasa en su ejecución y, en la mayor parte de los casos, en realidad se cuida a sí mismo. La naturaleza robusta del sistema de archivos de Linux ayuda en ese tipo de situaciones. Sin embargo, puede ser que le aparezca a usted este mensaje:

```
*** Ha ocurrido un error durante la verificación del sistema  
*** de archivos, haciendo que usted cayera en un shell;  
*** el sistema se reiniciará cuando usted salga del shell.
```

En este punto, necesita usted ejecutar **fsck** a mano y dar respuesta a sus mensajes por sí mismo.

Si encuentra que un sistema de archivos no se está comportando como debe (los mensajes del registro son un indicio excelente de este tipo de anomalía), puede ser que quiera ejecutar **fsck** por sí mismo en un sistema en funcionamiento. El único inconveniente es que el sistema de archivos en cuestión debe desmontarse para que esto funcione. Si decide seguir este camino, asegúrese de volver a montar ese sistema de archivos cuando haya terminado.

El nombre **fsck** no es el título apropiado para la herramienta de reparación de ext3; en realidad sólo es una envoltura. La envoltura **fsck** trata de determinar qué clase de sistema de archivos necesita repararse y, entonces, ejecuta la herramienta apropiada de reparación, pasando cualesquiera parámetros que se pasaron a **fsck**. En ext2, la herramienta real se llama **fsck.ext2**. Para el sistema de archivos ext3, la herramienta real es **fsck.ext3**; para el sistema de archivos VFAT, la herramienta es **fsck.vfat** y para un sistema de archivos ReiserFS, la utilidad se llama **fsck.reiserfs**. Por ejemplo, cuando se tiene una caída del sistema en una partición formateada según ext2, puede ser que necesite llamar **fsck.ext2** en forma directa, en lugar de apoyarse en otra aplicación para que la llame automáticamente en lugar de usted.

Por ejemplo, para ejecutar **fsck** en el sistema de archivos /dev/mapper/VolGroup00-LogVol02, montado en el directorio /home, ejecutará los comandos siguientes. Primero, para desmontar el sistema de archivos, teclee

```
[root@serverA ~]# umount /home
```

NOTA En el paso anterior, se supone que, en ese momento, el sistema de archivos /home no se está usando o no ha tenido un proceso acceso a él.

Puesto que sabemos que este sistema particular de archivos es ext3, podemos llamar directamente a la utilidad correcta (**fsck.ext3**) o, de manera escueta, usar la utilidad **fsck**. Teclee

```
[root@serverA ~]# fsck /dev/mapper/VolGroup00-LogVol02
fsck 1.35 (28-Feb-2024)
e2fsck 1.35 (28-Feb-2024)
/dev/mapper/VolGroup00-LogVol02: clean, 11/393216 files, 21703/786432 blocks
```

La salida anterior hace ver que el sistema de archivos está marcado como limpio. Para forzar a que se haga la verificación del sistema de archivos y se dé como respuesta sí a todas las preguntas, a pesar de lo que su OS piensa, teclee

```
[root@serverA ~]# fsck.ext3 -f -y /dev/mapper/VolGroup00-LogVol02
```

¿Qué hacer si todavía tenemos errores?

En principio, relájese. La verificación **fsck** rara vez encuentra problemas que no pueda resolver por sí misma. Cuando en realidad pide la intervención humana, a menudo basta con decirle a **fsck** que ejecute su sugerencia predeterminada. Muy rara vez sucede que con un solo paso de **e2fsck** no se eliminan todos los problemas.

En las raras ocasiones en que se necesite una segunda ejecución *no deben de* aparecer más errores. Si así sucede, lo más probable es que esté encarando una falla del hardware. Recuerde empezar con la comprobación obvia de energía eléctrica confiable y de cables bien conectados. Cualquier que esté haciendo funcionar sistemas SCSI debe verificar que está usando el tipo correcto de terminador, que los cables no son demasiado largos, que las SCSI ID no se encuentran en conflicto y que la calidad del cable es adecuada (la SCSI es en especial quisquillosa acerca de la calidad de los cables).

El directorio **lost+found**

Otra situación rara es cuando **fsck** encuentra segmentos de archivos que no pueden volverse a unir con el archivo original. En esos casos, colocará el fragmento en el directorio **lost+found** de la partición. Dicho directorio está ubicado en donde está montada la partición, de modo que si, por ejemplo, se monta `/dev/mapper/VolGroup00-LogVol02` en `/home`, entonces `/home/lost+found` se correlaciona con el directorio **lost+found** para ese sistema de archivos en particular.

Cualquier cosa puede ir hacia un directorio **lost+found**: fragmentos de archivo, directorios e, incluso, archivos especiales. Cuando archivos normales concluyen allí, debe estar agregado un propietario del archivo y puede ponerse en contacto con él y ver si necesita los datos (lo típico es que no sea así). Si encuentra un directorio en **lost+found**, lo más probable es que quiera intentar restablecerlo a partir de los respaldos más recientes, en lugar de tratar de reconstruirlo desde allí.

En el peor de los casos, **lost+found** le dice si algo se desubicó. De nuevo, tales errores son extraordinariamente raros.

ADICIÓN DE UN DISCO NUEVO

En Linux, el proceso de agregar un nuevo disco en la plataforma de Intel (`x86`) es más o menos fácil. Suponiendo que está agregando un disco que es de tipo semejante a los que tiene en existencia (por ejemplo, agregar un disco IDE a un sistema que ya cuenta con unidades IDE, o bien, agregar un disco SCSI a un sistema que ya tiene unidades SCSI), el sistema debe detectar en forma automática el nuevo disco en el momento de la inicialización. Todo lo que resta es formar las particiones y crear un sistema de archivos en él.

Si está agregando un nuevo tipo de disco (como un disco SCSI en un sistema que sólo tiene unidades IDE), puede ser que necesite asegurarse que su núcleo soporta el nuevo hardware. Este soporte puede estructurarse directamente en el núcleo o encontrarse disponible como un módulo que puede descargarse (controlador). Note que los núcleos de la mayor parte de las distribuciones de Linux vienen con soporte para muchos controladores SCSI populares pero, en ocasiones, el lector se topará con combinaciones de núcleo y hardware muy conflictivos, en especial con los nuevos tableros madre que tienen juegos de chips muy raros.

Una vez que el disco está en su lugar, sencillamente inicialice el sistema y esté listo para avanzar. Si no está seguro acerca de si el sistema puede ver el nuevo disco, ejecute el comando **dmesg** y vea si el controlador está cargado y pudo hallar su disco. Por ejemplo,

```
[root@serverA ~]# dmesg | less
```

Panorama general de las particiones

En beneficio de la claridad, y en el caso de que necesite saber qué es una partición y cómo funciona, hagamos un breve repaso de este tema. Todos los discos deben estar *partidos*. Las *particiones* dividen el disco en segmentos y cada uno de éstos actúa como un disco completo por sí mismo. Una vez que se llena una partición, no se puede derramar en forma automática sobre otra partición. Se pueden hacer varias cosas con un disco partido, como instalar un OS en una sola partición que se extienda sobre el disco completo, instalar varios OS en sus propias particiones separadas en lo que es común llamar configuración de “inicialización dual” y usar las diferentes particiones para separar y restringir ciertas funciones del sistema en sus propias áreas de trabajo. Esta última razón es en especial pertinente en un sistema de usuarios múltiples, en donde no debe permitirse

que el contenido de los directorios iniciales de los usuarios crezcan en exceso y perturben funciones importantes de OS.

Convenciones acerca del nombramiento de los discos y particiones

En Linux, a cada disco se le da su propio nombre de dispositivo. Los archivos de dispositivos se almacenan bajo el directorio `/dev`. Los discos IDE empiezan con el nombre `hdX`, en donde X puede variar desde la `a` hasta la `z`, representando cada letra un dispositivo físico. Por ejemplo, en un sistema sólo de IDE, con un disco duro y un CD-ROM, con los dos en la misma cadena IDE, el disco duro sería `/dev/hda` y el CD-ROM sería `/dev/hdb`. Los dispositivos de disco se crean automáticamente en el transcurso de la instalación del sistema.

Cuando se crean las particiones, se usan nuevos dispositivos. Éstos toman la forma de `/dev/hdXY`, en donde X es la letra del dispositivo (como se describió en el párrafo anterior) y Y es el número de partición. Por tanto, la primera partición en el disco `/dev/hda` es `/dev/hda1`, la segunda sería `/dev/hda2`, y así sucesivamente.

Los discos SCSI siguen el mismo esquema básico que el de los IDE, excepto que en lugar de empezar con `hd`, empiezan con `sd`. Por lo tanto, la primera partición en el primer disco SCSI sería `/dev/sda1`, la segunda partición en el tercer disco SCSI sería `/dev/sdc2`, etcétera.

ADMINISTRACIÓN DE VOLÚMENES

Puede ser que el lector haya advertido al principio que, en partes del texto, usamos los términos partición y volumen en forma intercambiable. Aunque no son exactamente lo mismo, los conceptos se trasladan. La administración de volúmenes es un nuevo enfoque para tratar con los discos y las particiones. En lugar de visualizar un disco o entidad de almacenamiento a lo largo de las fronteras de la partición, esas fronteras ya no están allí y ahora todo se ve como volúmenes.

¿Acaba de dar de vueltas su cabeza? No se preocupe si lo hizo: éste es un concepto difícil. Intentaremos analizar esto de nuevo con más detalle.

Este nuevo enfoque para tratar con las particiones se llama Logical Volume Management (LVM, Administración de volúmenes lógicos) en Linux. Se presta para obtener varios beneficios y elimina las restricciones, los apremios y las limitaciones que impone el concepto de particiones. Algunos de los beneficios son

- ▼ Mayor flexibilidad para las particiones del disco
- Fácil redimensionamiento en línea de los volúmenes
- Fácil aumento en el espacio de almacenamiento sencillamente al agregar nuevos discos a los recursos comunes
- ▲ Uso de tomas instantáneas

Los siguientes son algunos términos importantes de la administración de volúmenes.

Volumen físico (PV, physical volume) Lo típico es que esto se refiera al disco (o los discos) duro(s) físicos o a cualquier otra entidad física de almacenamiento, como un arreglo RAID del hardware o un dispositivo (o dispositivos) RAID de software. Puede haber una sola entidad de almacenamiento (por ejemplo, una partición) en un PV.

Grupo de volúmenes (VG, volume group) Los grupos de volúmenes se usan para alojar uno o más volúmenes físicos y volúmenes lógicos en una sola unidad administrativa. Un grupo de volúmenes

se crea de volúmenes físicos. Los VG son sencillamente una colección de PV: sin embargo, los VG no se pueden montar. Son más semejantes a los discos virtuales en blanco.

Volumen lógico (LV, *logical volume*) Éste quizá es el concepto de la LVM más difícil de captar, porque los volúmenes lógicos (LV) son los equivalentes a las particiones del disco en un mundo que no sea de LVM. El LV tiene la apariencia de un dispositivo estándar de bloques. En el LV es en el que ponemos los sistemas de archivos. Es el LV el que se monta. Es al LV al que se le aplica **fsck**, si es necesario.

Los LV se crean del espacio disponible en los VG. Para el administrador, un LV se ve como una partición contigua independiente de los PV reales que lo forman.

Extensiones Existen dos clases de extensiones: las físicas y las lógicas. Se dice que los volúmenes físicos (PV) están divididos en trozos o unidades de datos llamados “extensiones físicas”. Y se dice que los volúmenes lógicos están divididos en trozos o unidades de datos llamados “extensiones lógicas”.

Creación de particiones y volúmenes lógicos

En el curso del proceso de instalación, es probable que use una “bella” herramienta con una bonita entrada GUI, para crear particiones. Las herramientas GUI de las que se dispone entre las diversas distribuciones de Linux varían mucho en apariencia y facilidad de uso. Una herramienta que se puede usar con el fin de realizar la mayor parte de las tareas de efectuar las particiones y que tiene una apariencia y sensación unificadas, sin importar el sabor de Linux, es la venerable utilidad **fdisk**. Aunque es pequeña y algo embarazosa, es una herramienta confiable para realizar particiones. Además, en el caso que necesite detectar fallas en un sistema que se ha desempeñado en realidad mal, debe familiarizarse con las herramientas básicas, como **fdisk**. Otras utilidades muy poderosas de la línea de comandos para administrar particiones son **sfdisk** y **cfdisk**, así como la mucho más reciente utilidad **parted**; esta última es mucho más amigable para el usuario y tiene mucho más funcionalidades integradas que las de otras herramientas. De hecho, muchas de las herramientas GUI para administrar particiones llaman al programa **parted** en su final.

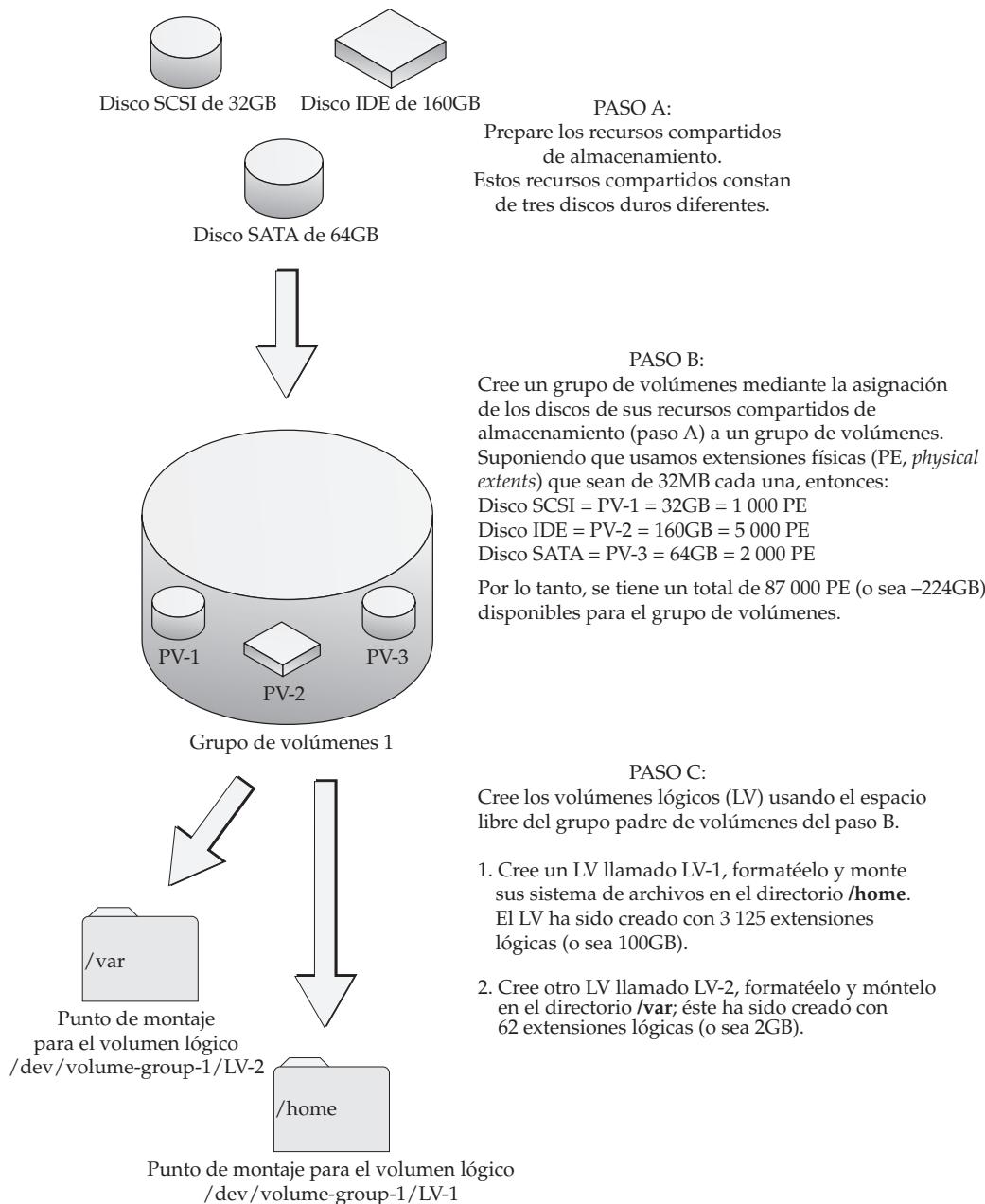
En el transcurso de la instalación del OS, como se cubrió en el capítulo 2, se le pidió que dejara algo de espacio libre, sin particiones. Ahora usaremos ese espacio libre para demostrar algunos conceptos de la LVM, recorriendo los pasos requeridos para crear un volumen lógico.

En particular, crearemos un volumen lógico que alojará el contenido de nuestro directorio **/var** actual. Debido a que en el curso de la instalación del OS *no* se creó un volumen “/var” separado, el contenido del directorio **/var** en la actualidad está almacenado bajo el volumen que contiene el árbol raíz (“/”). La idea general es que, debido a que, por lo común, el directorio **/var** se usa para contener datos que cambian y crecen con frecuencia (como los archivos de registro cronológico), resulta prudente poner su contenido en su propio sistema separado de archivos.

Los pasos relacionados con la creación de un volumen lógico se pueden resumir de esta manera:

1. Inicialice una partición común para que la use el sistema LVM (o sencillamente cree una partición del tipo Linux (0x8e)).
2. Cree volúmenes físicos con base en la partición del disco duro.
3. Asigne el volumen (o volúmenes) físico(s) al grupo (o a los grupos) de volúmenes.
4. Por último, cree volúmenes lógicos dentro de los grupos de volúmenes y asigne puntos de montaje para esos volúmenes lógicos, después de formatear.

En la ilustración que sigue, se muestra la relación entre los discos, los volúmenes físicos (PV), los grupos de volúmenes (VG) y los volúmenes lógicos (LV) en LVM:



PRECAUCIÓN El proceso de creación de particiones es irrevocablemente destructivo para los datos que ya están en el disco. Antes de crear, cambiar o eliminar particiones en cualquier disco, debe estar muy seguro de lo que está haciendo y de sus consecuencias.

La sección siguiente se dividirá en varias partes:

- ▼ Creación de una partición
- Creación de un volumen físico
- Asignación de un volumen físico a un grupo de volúmenes
- ▲ Creación de un volumen lógico

El proceso completo, desde el principio hasta el final, puede parecer un poco largo. En realidad, es un proceso muy sencillo en sí mismo, pero entremezclamos los pasos con algunos *pasos adicionales*, junto con algunas notas y explicaciones.

Empecemos el proceso. Note que, en la tabla 7-3, se da una lista de algunas de las utilidades LVM que estaremos usando en el desarrollo del proceso.

Comando LVM	Descripción
lvcreate	Usado para crear un nuevo volumen lógico en un grupo de volúmenes, mediante la asignación de extensiones lógicas de los recursos compartidos físicos libres de ese grupo de volúmenes.
lvdisplay	Presenta los atributos de un volumen lógico, como estado de lectura/escritura, tamaño e información de toma instantánea.
pvcREATE	Inicializa un volumen físico para usarse con el sistema LVM.
pvdisplay	Presenta los atributos de los volúmenes físicos, como tamaño y tamaño de la PE.
vgcreate	Usado para crear nuevos grupos de volúmenes a partir de dispositivos de bloques creados con el uso del comando pvcREATE .
vgextend	Usado para agregar uno o más volúmenes físicos a un grupo existente de volúmenes para extender su tamaño.
vgdisplay	Presenta los atributos de los grupos de volúmenes.

Tabla 7-3. Utilidades de LVM

Creación de una partición

Estaremos usando el espacio libre sin particiones en el disco principal del sistema, /dev/hda.

1. Empiece por ejecutar **fdisk** con el parámetro **-l** para hacer una lista de la tabla actual de particiones. Teclee

```
[root@serverA ~]# fdisk -l
Disk /dev/hda: 10.7 GB, 10737418240 bytes
... (OUTPUT TRUNCATED) ...
/dev/hda2          26        1200    9438187+  8e  Linux LVM
```

2. Enseguida, empecemos el proceso real de volver a realizar particiones, usando de nuevo **fdisk**. Teclee

```
[root@serverA ~]# fdisk /dev/hda
The number of cylinders for this disk is set to 1305.
... (OUTPUT TRUNCATED) ...
2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)
Command (m for help):
```

Se le presentará con un mensaje sencillo de **fdisk** "Command (m for help):" [Comando (m para obtener ayuda):]

3. Imprima una vez más la tabla de particiones mientras esté dentro del programa **fdisk**. Teclee **p** en mensaje de **fdisk** para imprimir la tabla de particiones.

```
Command (m for help): p
Disk /dev/hda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Device     Boot   Start      End      Blocks   Id  System
/dev/hda1    *       1       25     200781   83  Linux
/dev/hda2           26      1200    9438187+  8e  Linux LVM
```

Vale la pena hacer notar unos cuantos hechos referentes a esta salida:

- ▼ El tamaño total del disco es aproximadamente de 10.7GB.
- Se tienen en la actualidad dos particiones definidas en el sistema muestra. Las particiones son /dev/hda1 y /dev/hda2.
- La partición /dev/hda1 es del tipo "Linux" (0x83) y la /dev/hda2 es del tipo "Linux LVM" (0x8e).
- Con base en el esquema de particiones que elegimos en el curso de la instalación del OS, podemos deducir que en /dev/hda1 se aloja el sistema de archivos /boot y en /dev/hda2 se aloja todo lo demás (como referencia, vea la salida del comando **df**).
- El disco completo se extiende por 1 305 cilindros.
- ▲ La última partición; es decir, la /dev/hda2, finaliza en la frontera del cilindro 1 200. Por lo tanto, se tiene lugar para crear una partición que ocupará el espacio desde el cilindro 1 201 hasta el último cilindro en el disco (es decir, el 1 305).

4. Teclee **n** en el mensaje para crear una nueva partición.

```
Command (m for help) : n
```

NOTA Si tiene curiosidad acerca de las otras cosas que puede hacer en el mensaje de **fdisk**, teclee **m** para presentar un menú de ayuda.

5. Teclee **p** para seleccionar un tipo de partición primaria.

```
e      extended  
p      primary partition (1-4)  
p
```

6. Queremos crear la tercera partición primaria, Teclee **3** cuando se le pida un número de partición:

```
Partition number (1-4) : 3
```

7. El paso siguiente es especificar el tamaño de la partición. En primer lugar, elegimos el límite inferior. Acepte el valor predeterminado para el primer cilindro. Teclee **1201**.

```
First cylinder (1201-1305, default 1201) : 1201
```

8. En lugar de designar un valor en megabytes para el tamaño de esta partición, introducimos el número del último cilindro, tomando de este modo el resto del disco. Acepte el número predeterminado que se sugiere para el último cilindro. En nuestro sistema de ejemplo, este valor es 1 305. Teclee **1305**.

```
Last cylinder or +size or +sizeM or +sizeK (1201-1305, default 1305) : 1305
```

9. De manera predeterminada, **fdisk** crea particiones del tipo ext2 (es decir, 0x83). Pero queremos crear una partición del tipo "Linux LVM". Cambie el tipo de partición de la Linux predeterminada (0x83) al tipo "Linux LVM". Para hacer esto, usamos el comando **t** (cambiar el tipo de partición). Teclee **t**.

```
Command (m for help) : t
```

10. Introduzca el número de la partición cuyo tipo quiere cambiar. Queremos cambiar el tipo para la partición /dev/hda3 que se acaba de crear, de modo que teclee **3** cuando se le pida un número de partición.

```
Partition number (1-4) : 3
```

11. Introduzca el tipo de partición para "Linux LVM". Teclee **8e** en el mensaje:

```
Hex code (type L to list codes) : 8e
```

NOTA Puede obtener la lista de los códigos hexadecimales para los tipos de partición de los que se dispone al teclear **L**.

12. Vea los cambios que ha hecho al ver la tabla de particiones. Teclee **p**.

```
Command (m for help): p
Disk /dev/hda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Device     Boot   Start     End   Blocks   Id  System
/dev/hda1    *       1      25   200781   83  Linux
/dev/hda2          26     1200  9438187+   8e  Linux LVM
/dev/hda3        1201    1305  843412+   8e  Linux LVM
```

13. Una vez que esté satisfecho con sus cambios, consigne o escriba los cambios que ha hecho en la tabla de particiones del disco, usando el comando **w** (escribir tabla en el disco):

```
Command (m for help): w
```

14. Abandone la utilidad **fdisk**. Teclee **q**.

```
Command (m for help): q
```

15. Cuando haya regresado al mensaje del shell, reinicie el sistema para dejar que el núcleo de Linux reconozca de manera apropiada la nueva tabla de particiones. Teclee

```
[root@serverA ~]# reboot
```

Creación de un volumen físico

A continuación, cree el propio volumen físico.

- Después de que el sistema regresa del reinicio, vuelva a entrar como el superusuario.
- En primer lugar, veamos los volúmenes físicos definidos actualmente en el sistema. Teclee

```
[root@serverA ~]# pvdisplay
--- Physical volume ---
PV Name           /dev/hda2
VG Name           VolGroup00
PV Size          9.00 GB / not usable 0
... (OUTPUT TRUNCATED) ...
```

Tome nota del campo de nombre del volumen físico (PV Name).

- Use el comando **pvcREATE** para inicializar la partición que creamos al principio como un volumen físico. Teclee

```
[root@serverA ~]# pvcREATE /dev/hda3
Physical volume "/dev/hda3" successfully created
```

- Use el comando **pvdisplay** para ver sus cambios una vez más. Teclee

```
[root@serverA ~]# pvdisplay
--- Physical volume ---
PV Name           /dev/hda2
VG Name           VolGroup00
... (OUTPUT TRUNCATED) ...
```

```
--- NEW Physical volume ---
PV Name          /dev/hda3
VG Name
PV Size         823.46 MB
... (OUTPUT TRUNCATED) ...
```

Asignación de un volumen físico a un grupo de volúmenes

En esta sección, asignaremos el volumen físico creado al principio a un grupo de volúmenes (VG).

1. Primero, use el comando **vgdisplay** para ver los grupos actuales de volúmenes que podrían existir en su sistema. Teclee

```
[root@serverA ~]# vgdisplay
--- Volume group ---
VG Name          VolGroup00
Format           lvm2
... (Output truncated) ...
VG Size          9.00 GB
PE Size          32.00 MB
Total PE        288
Alloc PE / Size 286 / 8.94 GB
Free PE / Size  2 / 64.00 MB
VG UUID          JgPahd-1TBY-L5sT-tqho-KCk6-HxTy-rNCdz8
```

De la salida anterior, podemos decir que

- ▼ El nombre del grupo de volúmenes (VG Name) es VolGroup00.
 - El tamaño actual del VG es de 9.00GB (éste debe aumentar en el momento en que terminemos).
 - El tamaño de la extensión física es de 32MB y se tienen un total de 288 PE.
 - ▲ Sólo hay dos extensiones físicas que se encuentran libres en el VG. Son equivalentes a un espacio de 64MB.
2. Asigne el PV al grupo de volúmenes, usando el comando **vgextend**. La sintaxis para el comando es
- ```
Vgextend [options] VolumeGroupName PhysicalDevicePath
```
- Sustituyendo los valores correctos en este comando, teclee
- ```
[root@serverA ~]# vgextend VolGroup00 /dev/hda3
Volume group "VolGroup00" successfully extended
```
3. Vea sus cambios con el comando **vgdisplay**. Teclee

```
[root@serverA ~]# vgdisplay
--- Volume group ---
VG Name          VolGroup00
... (Output truncated) ...
```

```

Act PV          2
VG Size        9.78 GB
PE Size        32.00 MB
Total PE       313
Alloc PE / Size 286 / 8.94 GB
Free PE / Size 27 / 864.00 MB

```

Note que los valores del VG Size, Total PE y FreePE han aumentado en forma considerable. Ahora tenemos un total de 27 PE libres (o sea 864MB).

Creación de un volumen lógico (LV)

Ahora que tenemos algo de espacio en el VG, podemos seguir adelante y crear el volumen lógico (LV) final.

1. Primero, veamos los LV actuales en el sistema. Teclee

```

[root@serverA ~]# lvdisplay
--- Logical volume ---
  LV Name        /dev/VolGroup00/LogVol00
  VG Name        VolGroup00
... (Output truncated) ...
--- Logical volume ---
  LV Name        /dev/VolGroup00/LogVol02
  VG Name        VolGroup00
... (Output truncated) ...
--- Logical volume ---
  LV Name        /dev/VolGroup00/LogVol03
  VG Name        VolGroup00
... (Output truncated) ...
--- Logical volume ---
  LV Name        /dev/VolGroup00/LogVol01
  VG Name        VolGroup00

```

La salida anterior muestra los LV actuales: /dev/VolGroup00/LogVol00, /dev/VolGroup00/LogVol02, /dev/VolGroup00/LogVol03, etcétera.

2. Con la información básica con la que ahora contamos, crearemos un LV usando la misma convención de nombramiento que se está usando en la actualidad en el sistema. Crearemos un cuarto LV llamado "LogVol04". La trayectoria completa hasta el LV /dev/VolGroup00/LogVol04. Teclee.

```

[root@serverA ~]# lvcreate -l 27 --name LogVol04 VolGroup00
... (OUTPUT TRUNCATED) ...
Logical volume "LogVol04" created

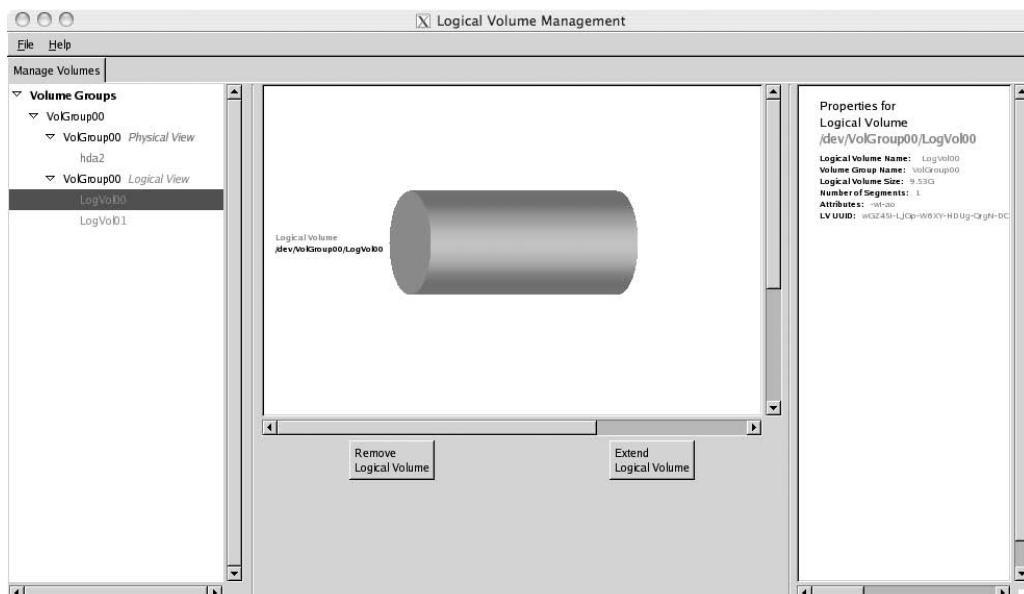
```

NOTA En realidad, puede nombrar su LV de la manera que quiera. Sólo le dimos al nuestro el nombre de **LogVol04** para ser coherentes. Pudimos haber reemplazado LogVol04 con otro nombre como “mi-volumen”, si lo hubiéramos querido. El valor para las opciones **--name (-n)** determina el nombre para el LV. Con la opción **-l** se especifica el tamaño en unidades de extensiones físicas (vea el paso 1 en “Asignación de un volumen físico a un grupo de volúmenes”). También pudimos haber especificado el tamaño en megabytes, usando una opción como **-L 864M**.

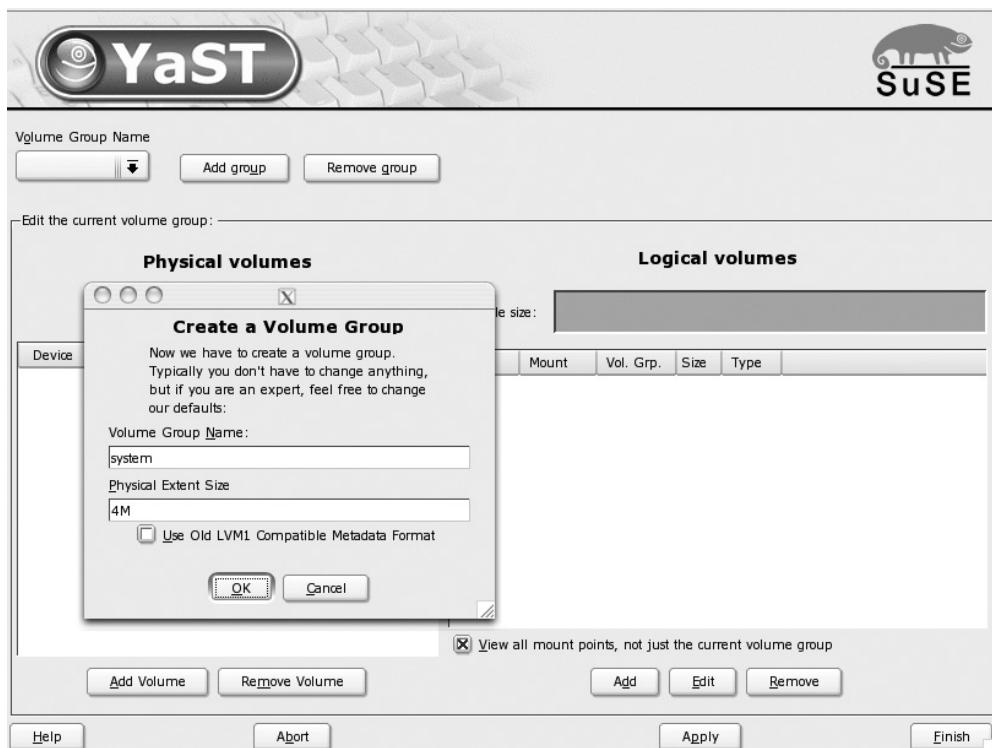
3. Vea el LV que creó. Teclee

```
[root@serverA ~]# lvdisplay /dev/VolGroup00/LogVol04
--- Logical volume ---
  LV Name           /dev/VolGroup00/LogVol04
  VG Name           VolGroup00
  ... (Output truncated) ...
  LV Size           864.00 MB
  Current LE        27
```

Las distribuciones Fedora y RHEL de Linux tienen una herramienta GUI que puede simplificar mucho toda la administración de un sistema LVM. El comando **system-config-lvm** lanzará la herramienta, como se muestra enseguida:



SuSE Linux también tiene una herramienta GUI muy capaz para administrar discos, particiones y la LVM. Emite el comando **yast2 lvm_config** para lanzar la utilidad que a continuación se muestra:



CREACIÓN DE LOS SISTEMAS DE ARCHIVOS

Con los volúmenes creados, necesita poner sistemas de archivos en ellos (si está acostumbrado a Windows de Microsoft, esto es semejante a formatear el disco una vez que ha establecido las particiones en él).

El tipo de sistema de archivos que quiere crear determinará la utilidad particular que debe usar. En este proyecto, queremos crear un sistema de archivos tipo ext3; por lo tanto, usaremos la utilidad **mkfs.ext3**. Se dispone de muchos parámetros de la línea de comandos para la herramienta **mkfs.ext3** pero, en lo que sigue, la usaremos en su forma más sencilla.

Los que siguen son los pasos para crear un sistema de archivos:

1. El único parámetro de la línea de comandos que por lo común tendrá que especificar es el nombre de la partición (o volumen) sobre la cual debe ir el sistema de archivos. Para crear un sistema de archivos, en el `/dev/VolGroup00/LogVol04`, debe emitir el comando que sigue:

```
[root@serverA ~]# mkfs.ext3 /dev/VolGroup00/LogVol04
mke2fs 1.35 (28-Feb-2004)
max_blocks 226492416, rsv_groups = 6912, rsv_gdb = 53
OS type: Linux
Block size=4096 (log=2)
... (Output truncated)...
110656 inodes, 221184 blocks
Superblock backups stored on blocks:
    32768, 98304, 163840
Writing inode tables: done
inode.i_blocks = 1704, i_size = 4243456
Creating journal (4096 blocks): done
Writing superblocks and file system accounting information: done
This file system will be automatically checked every 38 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override
```

Una vez que el comando anterior se ejecuta hasta completarse, el lector ha terminado con la creación del sistema de archivos. A continuación, empezaremos el proceso para tratar de restablecer el contenido del sistema de archivos /var.

2. Cree una carpeta temporal que usará como el punto de montaje para el nuevo sistema de archivos. Créela bajo la carpeta raíz. Teclee

```
[root@serverA ~]# mkdir /new_var
```

3. Monte el volumen lógico LogVol04 en el directorio /new_var. Teclee

```
[root@serverA ~]# mount /dev/VolGroup00/LogVol04 /new_var
```

4. Copie el contenido del directorio /var en el directorio /new_var. Teclee

```
[root@serverA ~]# cp -rp /var/* /new_var/
```

5. Para evitar llevar el sistema hacia el modo de un solo usuario con el fin de realizar los sensibles pasos que siguen, tenemos que usar unos cuantos viejos trucos militares. Teclee

```
[root@serverA ~]# mount --bind /var/lib/nfs/rpc_pipefs \
/new_var/lib/nfs/rpc_pipefs
```

El paso anterior es necesario porque el sistema de pseudoarchivos rpc_pipefs tiene que ser montado bajo una subcarpeta en el directorio /var.

6. Ahora puede dar un nombre nuevo al /var como /old_var. Teclee

```
[root@serverA ~]# mv /var /old_var
```

7. Cree un directorio /var nuevo y vacío. Teclee

```
[root@serverA ~]# mkdir /var
```

8. Restablezca los contextos de seguridad para la nueva carpeta /var de modo que los demonios que necesita puedan usarla. Teclee

```
[root@serverA /]# restorecon -R /var
```

NOTA El paso anterior sólo es necesario en un sistema que esté ejecutando un núcleo activado por SELinux, como Fedora o RHEL.

Ahora, casi hemos terminado. Necesitamos crear una entrada para el nuevo sistema de archivos en el archivo `/etc/fstab`. Para hacerlo, debemos editar este último archivo de modo que nuestros cambios puedan tener efecto la próxima vez que el sistema se reinicie. Abra el archivo para editar, con cualquier editor de textos de su agrado, y agregue la entrada siguiente en él:

```
/dev/VolGroup00/LogVol04 /var ext3 defaults 1 2
```

SUGERENCIA También puede usar el comando `echo` para añadir el texto antes dado al final del archivo. El comando es

```
echo "/dev/VolGroup00/LogVol04 /var ext3 defaults 1 2" >> /etc/fstab.
```

9. Éste será un buen momento para reiniciar el sistema. Teclee
- [root@serverA /]# `reboot`
10. Esperemos que el sistema retorne bien. Después que el sistema inicializa, borre las carpetas `/old_var` y `/new_var` usando el comando `rm`.

NOTA Si, durante la inicialización del sistema, el proceso de arranque fue lento en especial al iniciar el sistema “logger service”, no se preocupe demasiado; en algún momento hará una pausa y continuará con el proceso de inicialización. Pero necesitará fijar los contextos apropiados de seguridad para los archivos que están ahora bajo la carpeta `/var`, al ejecutar de nuevo el comando `restorecon -R /var`, con los archivos actuales que están ahora en el directorio. Y, después, reinicie el sistema una vez más.

RESUMEN

En este capítulo, cubrimos el proceso de administración de sus sistemas de archivos, desde la creación de particiones hasta la creación de volúmenes físicos, pasando por la extensión de un grupo existente de volúmenes y, después, creando el volumen lógico final. También recorrimos el proceso de llevar un directorio muy sensible del sistema sobre su propio sistema separado de archivos. En el ejercicio, se detalló lo que podría necesitar para hacerlo mientras está administrando un servidor Linux en el mundo real. Con esta información. Está usted armado con lo que necesita para administrar un servidor Linux comercial, en diversos entornos.

Como cualquier sistema operativo, Linux pasa por cambios de tiempo en tiempo. Aunque los diseñadores y mantenedores de los sistemas de archivos consumen grandes cantidades de tiempo para conservar igual la interfaz, encontrará que algunas alteraciones se asomarán de vez en vez. Algunas veces serán simplificaciones de la interfaz, en otras, serán mejoras de gran importancia en el propio sistema de archivos. Mantenga sus ojos abiertos para estos cambios. Linux proporciona un espléndido sistema de archivos que es robusto, que responde y que, en general, resulta un placer usar. Tome las herramientas que hemos discutido en este capítulo y averígüelo usted mismo.

CAPÍTULO 8



Servicios centrales
del sistema

Sin importar la distribución, la configuración de la red y el diseño global del sistema, cada sistema Linux se embarca con algunos servicios centrales. Algunos de estos servicios incluyen **init**, **syslogd**, **cron** y otros. Las funciones que desempeñan estos servicios pueden ser simples, pero también son fundamentales. Sin su presencia, se perdería gran parte del poder de Linux.

En este capítulo, discutiremos cada uno de los servicios centrales, además de otro servicio muy útil del sistema llamado **xinetd**. También analizaremos el archivo de configuración correspondiente de cada servicio y el método sugerido de despliegue (si resulta apropiado). El lector encontrará que las secciones que cubren estos sencillos servicios no son terriblemente largas, pero no descuide este material. Recomendamos con vehemencia que se tome algo de tiempo para familiarizarse con sus implicaciones. Se han realizado muchas soluciones creativas a través del uso de estos servicios. Tenemos la esperanza de que este capítulo inspirará unas cuantas más.

EL SERVICIO INIT

El proceso **init** es el patrón de todos los procesos. *Siempre* como el primer proceso que se inicia en cualquier sistema basado en UNIX (como Linux), la ID del proceso **init** es 1. Si fallara alguna vez **init**, lo más probable es que el resto del sistema hará lo mismo.

NOTA Si se quiere ser estrictamente correcto desde el punto de vista técnico, **init** en realidad no es el mismo primer proceso que se ejecuta. Pero para permanecer correctos desde el punto de vista político, ¡supondremos que lo es! El lector también debe tener presente que algunos llamados sistemas Linux endurecidos respecto a la seguridad de manera deliberada hacen aleatoria la PID de **init**, de modo que no se sorprenda si alguna vez se encuentra en uno de esos sistemas y advierte que la PID de **init** *no* es uno (1).

El proceso **init** desempeña dos papeles. El primero es ser el último proceso padre. Debido a que **init** nunca muere, el sistema siempre puede estar seguro de su presencia y, si es necesario, hacer referencia a él. La necesidad de referirse a **init** suele suceder cuando un proceso muere antes que todos sus procesos hijos generados se hayan completado. Esto hace que los hijos hereden a **init** como su proceso padre. Una ejecución rápida del comando **ps -af** mostrará varios procesos que tendrán una ID del proceso padre (PPID, *parent process ID*) de 1.

La segunda tarea para **init** es manejar los diversos niveles de ejecución para que se ejecuten los programas apropiados cuando se alcanza un nivel particular de ejecución. Este comportamiento se define por medio del archivo **/etc/inittab**.

El archivo /etc/inittab

El archivo **/etc/inittab** contiene toda la información que **init** necesita para iniciar los niveles de ejecución. El formato de cada línea en este archivo es como la que sigue:

id:runlevels:action:process

SUGERENCIA Las líneas que empiezan con el símbolo de número (#) son comentarios. Eche un vistazo a su propio /etc/inittab y encontrará que ya está comentado en forma liberal. Si alguna vez en realidad necesita hacer un cambio a este archivo, hágase un favor incluyendo comentarios liberales para explicar lo que ha hecho.

En la tabla 8-1 se explica el significado de cada uno de los cuatro campos de una entrada en el archivo /etc/inittab, en tanto que en la 8-2, se definen algunas opciones comunes de las que se dispone para el campo **action** de este archivo.

Ahora echemos una mirada a una entrada muestra de un archivo /etc/inittab:

```
# Si la energía eléctrica se restablece antes de que se inicie el paro, cancelle éste.  
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"
```

En este caso:

- ▼ La primera línea, la cual empieza con el signo de número (#), es una entrada de comentario y se ignora.
- **pr** es el identificador único.
- 1, 2, 3, 4 y 5 son los niveles de ejecución en los cuales se puede activar este proceso.
- **powerokwait** es la condición con la cual se ejecuta el proceso.
- ▲ El comando **/sbin/shutdown**...es el proceso.

Ítem de /etc/inittab	Descripción
<i>id</i>	Una sucesión única de caracteres del 1 al 4 que identifica esta entrada en el archivo /etc/inittab.
<i>runlevels</i>	Los niveles de ejecución en los cuales el proceso debe llamarse. Algunos eventos son suficientemente especiales que pueden atraparse en todos los niveles de ejecución (por ejemplo, la combinación de teclas CTRL-ALT-DEL para reiniciar). Para indicar que un evento es aplicable a todos los niveles de ejecución, deje <i>runlevels</i> en blanco. Si quiere que ocurra algo en múltiples niveles de ejecución, sencillamente haga una lista de todos ellos en este campo. Por ejemplo, la entrada 123 en <i>runlevels</i> especifica algo que se ejecuta en los niveles 1, 2 o 3.
<i>action</i>	Describe cuál acción se debe tomar. En la tabla que sigue, se explican las opciones para este campo.
<i>process</i>	Nombra el proceso (o programa) para ejecutarse cuando se introduce el nivel de ejecución.

Tabla 8-1. Entradas de /etc/inittab

Campo <i>action</i> en /etc/inittab	Descripción
respawn	El proceso se reiniciará siempre que termine
wait	El proceso se reiniciará una vez cuando se introduzca el nivel de ejecución e init esperará para su terminación.
once	El proceso se reiniciará una vez cuando se introduzca el nivel de ejecución; sin embargo, init no esperará para la terminación antes posiblemente de ejecutar programas adicionales que deben ejecutarse en ese nivel particular.
boot	El proceso se ejecutará en la inicialización del sistema. En este caso, se ignora el campo de runlevels .
bootwait	El proceso se ejecutará en la inicialización del sistema e init esperará para la terminación de esa inicialización, antes de avanzar al siguiente proceso que va a ejecutarse.
ondemand	El proceso se ejecutará cuando se presente la solicitud de un nivel específico de ejecución (estos niveles son a , b y c). No se tienen cambios en el nivel de ejecución.
initdefault	Especifica el nivel predeterminado de ejecución para init en el arranque. Si no se especifica el predeterminado, al usuario se le pedirá un nivel en la consola.
sysinit	El proceso se ejecutará en el transcurso de la inicialización del sistema, antes de cualquiera de las entradas boot o bootwait .
powerwait	Si init recibe una señal, de otro proceso, de que hay problemas con la energía eléctrica, se ejecutará este proceso. Antes de continuar, init esperará para que este proceso finalice.
powerfail	Igual que powerwait , excepto que init no esperará a que el proceso finalice.

Tabla 8-2. Opciones disponibles para el campo *action* en el archivo /etc/inittab

Campo <code>action</code> en <code>/etc/inittab</code>	Descripción
<code>powerokwait</code>	Este proceso se ejecutará tan pronto como <code>init</code> esté informado que la energía eléctrica se ha restablecido.
<code>ctrlaltdel</code>	El proceso se ejecutará cuando <code>init</code> recibe una señal que indique que el usuario ha presionado la combinación de teclas CTRL-ALT-DELETE. Tenga presente que la mayor parte de los servidores X Window System capturan esta combinación de teclas y, por consiguiente, puede ser que <code>init</code> no reciba esta señal si X Window System está activo.

Tabla 8-2. Opciones disponibles para el campo `action` en el archivo `/etc/inittab` (*cont.*)

El comando `telinit`

Ha llegado el momento de confesar: la misteriosa fuerza que en realidad le dice a `init` cuándo cambiar los niveles de ejecución es el comando `telinit`. Este comando admite dos parámetros de la línea de comandos. Uno es el nivel deseado de ejecución acerca del cual `init` necesita saber y el otro es `-t sec`, en donde `sec` es el número de segundos que hay que esperar antes de decirle algo a `init`.



NOTA Si `init` en realidad cambia los niveles de ejecución es decisión de él. Es obvio que, por lo común, lo hace, o este comando no sería terriblemente útil.

Es mucho muy raro que usted tenga que ejecutar alguna vez el comando `telinit` por sí mismo. Por lo general, todo esto lo manejan para usted los scripts de arranque y paro.



NOTA Bajo la mayor parte de las implementaciones de UNIX (incluyendo Linux), el comando `telinit` en realidad sólo es un vínculo simbólico hacia el programa `init`. Debido a esto, algunos muchachos prefieren ejecutar `init` con el nivel de ejecución que ellos quieren, en lugar de usar `telinit`.

XINETD E INETD

Los programas `xinetd` e `inetd` son dos servicios populares en los sistemas Linux; `xinetd` es la encarnación más moderna del `inetd` más antiguo. Hablando de manera estricta, un sistema Linux puede funcionar de manera eficaz sin la presencia de ninguno de ellos. Pero algunos demonios se apoyan únicamente en la funcionalidad que proporcionan. De modo que si necesita `xinetd` o `inetd`, entonces lo necesita y no hay dos maneras acerca de ello.

Los programas **inetd** y **xinetd** son procesos demonios. Es probable que el lector sepa que los demonios son programas especiales que, después de iniciarse, de manera voluntaria se liberan del control de la terminal en la cual se iniciaron. El mecanismo principal mediante el cual los demonios pueden establecer la interfaz con el resto del sistema es a través de los canales de comunicación interprocesos (IPC, *interprocess communication channels*), enviando mensajes al archivo de registro cronológico de todo el sistema, o bien, anexándose a un archivo en el disco.

El papel de **inetd** es funcionar como un “superservidor” para otros procesos de la red relacionados con el servidor, como **telnet**, **ftp**, **tftp**, etcétera.

Es una filosofía sencilla: no a todos los procesos del servidor (incluyendo los que aceptan nuevas conexiones) se les llama con tanta frecuencia como para que requieran que un programa se esté ejecutando en la memoria todo el tiempo. La razón principal para la existencia de un superservidor es conservar los recursos del sistema. De modo que, en lugar de mantener de manera constante potencialmente docenas de servicios cargados en la memoria, esperando que sean usados, se hace una lista de todos en el archivo de configuración de **inetd**, **/etc/inetd.conf**. En su nombre **inetd** escucha las conexiones entrantes. De este modo, sólo es necesario que un proceso esté en la memoria.

Un beneficio secundario de **inetd** recae en aquellos procesos que necesitan conectividad con la red pero cuyos programadores no quieren tener que escribirlo en el sistema. El programa **inetd** manejará el código de la red y pasará las corrientes entrantes de ésta al proceso como su entrada estándar (**stdin**). Cualquier salida del proceso (**stdout**) se envía de regreso al anfitrión que ha conectado al proceso.

NOTA A menos que esté usted programando, no tiene que estar interesado en la característica **stdin**/**stdout** de **inetd**. Por otra parte, para alguien que quiere escribir un script sencillo y ponerlo a disposición a través de la red, vale la pena examinar esta muy poderosa herramienta.

Como regla empírica general, los servicios de bajo volumen (como **tftp**) suelen ejecutarse de la mejor manera a través del **inetd**, en tanto que los de volumen más elevado (como los servidores Web) se ejecutan mejor como un proceso solo que siempre está en la memoria listo para manejar solicitudes.

Las versiones actuales de Fedora, RHEL, SuSE, Mandrake e, incluso, Mac OS X se embarcan con una personificación más reciente de **inetd**, conocida como **xinetd**; el nombre es un acrónimo de “extended Internet services daemon” (demonio ampliado para servicios de Internet). El programa **xinetd** realiza la misma tarea que el **inetd** normal: ayuda a iniciar programas que proporcionan servicios de Internet. En lugar de hacer que esos programas se arranquen de manera automática en el curso de la inicialización del sistema y permanezcan sin usarse hasta que llegue una solicitud de conexión, **xinetd** permanece en el hueco correspondiente a esos programas y escucha en los puertos normales de servicio de éstos. Como resultado, cuando **xinetd** escucha una solicitud de servicio que corresponde a uno de los servicios que administra, entonces inicia o menosprecia el servicio apropiado.

En vista de que, por lo que respecta a la función, **xinetd** es semejante a **inetd**, se debe hacer notar que incluye un nuevo formato de archivo de configuración y gran cantidad de características adicionales. En el demonio **xinetd** se usa un formato de archivo de configuración que es muy diferente del clásico de **inetd**. (En la mayor parte de otras variantes de UNIX, incluyendo Solaris y FreeBSD, se usa el formato clásico de **inetd**.) Esto significa que si tiene una aplicación que se apoya en **inetd**, puede ser que necesite suministrar algunos ajustes manuales para hacerlo funcionar. Por supuesto, definitivamente debe ponerse en contacto con los desarrolladores de la aplicación

y hacerles saber del cambio, de modo que puedan liberar una versión más reciente que también funcione con el nuevo formato de configuración de **xinetd**.

En esta sección, cubriremos el nuevo demonio **xinetd**. Si en su sistema se usa **inetd**, debe poder ver el archivo **/etc/inetd.conf** y darse cuenta de las semejanzas entre **inetd** y **xinetd**.

El archivo **/etc/xinetd.conf**

El archivo **/etc/xinetd.conf** consta de una serie de bloques que toman el formato siguiente:

```
blockname
{
    variable = value
}
```

en donde **blockname** es el nombre del bloque que se está definiendo, **variable** es el nombre de una variable que se está definiendo dentro del contexto del bloque y **value** es el valor asignado a la **variable**. Cada bloque puede tener múltiples variables definidas dentro de él.

Un bloque especial se llama **defaults**. Cualesquiera que sean las variables definidas dentro de este bloque, se aplican a todos los demás bloques que se definan en el archivo.

Una excepción al formato del bloque es la instrucción **includedir**, la cual indica a **xinetd** que lea todos los archivos en el directorio y los considere parte del archivo **/etc/xinetd.conf** file.

Cualquier línea que empiece con un signo de número (#) es el comienzo de un comentario. El archivo en existencia **/etc/xinetd.conf** que se embarca con Fedora Core se mira como esto:

```
# Archivo simple de configuración para xinetd
# Algunos valores predeterminados e incluyen /etc/xinetd.d/

defaults
{
    instances          = 60
    log_type           = SYSLOG authpriv
    log_on_success     = HOST PID
    log_on_failure     = HOST
    cps                = 25 30
}

includedir /etc/xinetd.d
```

No se preocupe si todavía no le son familiares todas las variables y todos los valores; pasaremos por éas dentro de un momento. En primer lugar, asegúémonos que comprende el formato del archivo.

En este ejemplo, las dos primeras líneas del archivo son comentarios que explican lo que es el archivo y lo que hace. Después de los comentarios, ve el primer bloque: **defaults**. La primera variable que se define en este bloque es **instances** que se fija en el valor de 60. En este bloque, se definen cinco variables en total, la última es **cps**. Ya que este bloque se titula **defaults**, las variables que se fijen dentro de él se aplicarán a todos los bloques que se definan en el futuro. Por último, en la última línea del archivo se especifica que debe examinarse el directorio **/etc/xinetd.d** en relación con otros archivos que contienen más información de configuración. Esto hará que **xinetd** lea todos los archivos de ese directorio y los analice como si fueran parte del archivo **/etc/xinetd.conf**.

Variables y sus significados

En la tabla 8-3, se da una lista de algunos de los nombres de variables que son soportadas en el archivo `/etc/xinetd.conf`.

Al definir un servicio, no necesita especificar todas las variables. Las únicas requeridas son

- ▼ `socket_type`
- `user`
- `server`
- ▲ `wait`

Variable	Descripción
<code>id</code>	Este atributo se usa para especificar de manera única un servicio. Esto es útil porque existen servicios en los que se pueden usar protocolos diferentes y es necesario describirlos con entradas distintas en el archivo de configuración. De modo predeterminado, la ID es la misma que el nombre del servicio.
<code>type</code>	Se puede usar cualquier combinación de los valores siguientes: RPC si éste es un servicio RPC; INTERNAL , si <code>xinetd</code> suministra este servicio; o bien UNLISTED si éste es un servicio que no aparece en la lista del archivo <code>/etc/services</code> .
<code>disable</code>	Éste es el valor yes o no . Un valor de yes significa que, aun cuando el servicio esté definido, no está disponible para ser usado.
<code>socket_type</code>	Los valores válidos para esta variable son stream para indicar que este servicio es uno basado en flujo de datos, dgram para indicar que este servicio es un datagrama, o bien raw para indicar que en este servicio se usan datagramas IP en bruto. El valor stream conexión (TCP) (por ejemplo, Telnet y FTP). El valor dgram se refiere a corrientes de datagramas (UDP) (por ejemplo, el servicio TFTP es un protocolo basado en datagramas). En realidad existen otros protocolos fuera del alcance de TCP/IP; sin embargo, rara vez los encontrará.
<code>protocol</code>	Determina el tipo de protocolo (tcp o udp) para el tipo de conexión.
<code>wait</code>	Si ésta se fija en yes , sólo se procesará una conexión a la vez. Si ésta se fija en no , se permitirán conexiones múltiples al ejecutarse el demonio apropiado de servicios múltiples veces.

Tabla 8-3. Variables del archivo de configuración de xinetd

Variable	Descripción
user	Especifica el nombre del usuario bajo el cual se ejecutará este servicio. El nombre de usuario debe existir en el archivo <code>/etc/passwd</code> .
group	Especifica el nombre del grupo bajo el cual se ejecutará este servicio. El grupo debe existir en el archivo <code>/etc/group</code> .
instances	Especifica el número máximo de conexiones concurrentes que se permite manejar a este servicio. El valor predeterminado es sin límite, si la variable <code>wait</code> se fija en <code>nowait</code> .
server	El nombre del programa por ejecutar cuando se conecta este servicio.
server_args	Los argumentos pasados al servidor. Contrastando con <code>inetd</code> , el nombre del servidor no debe incluirse en <code>server_args</code> .
only_from	Especifica las redes de las cuales puede llegar una conexión válida (esta es la funcionalidad TCP Wrappers integrada). Usted puede especificar esto de tres maneras: como una dirección numérica, un nombre de anfitrión o una dirección de red con netmask. La dirección numérica puede tomar la forma de una dirección IP completa para indicar un anfitrión específico (como 192.168.1.1). Sin embargo, si cualquiera de los octetos finales son ceros, la dirección se tratará como una red en donde todos los octetos que son cero son comodines (por ejemplo, 192.168.1.0 significa cualquier anfitrión que empiece con los números 192.168.1). De modo alternativo, puede especificar el número de bits en la netmask después de una barra diagonal (por ejemplo, 192.168.1.0/24 significa la dirección de una red de 192.168.1.0 con una netmask de 255.255.255.0).
no_access	Lo opuesto a <code>only_from</code> en el sentido de que, en lugar de especificar las direcciones desde las cuales una conexión es válida, en esta variable se especifican las direcciones desde las cuales una conexión es inválida. Puede tomar el mismo tipo de parámetros que los de <code>only_from</code> .
log_type	Determina a dónde irá la información de registro cronológico para ese servicio. Se tienen dos valores válidos: SYSLOG y FILE . Si se especifica SYSLOG , también debe especificar en cuál recurso de <code>syslog</code> se va hacer el registro (para obtener más información sobre los recursos, vea “El demonio <code>syslogd</code> ”, más adelante en este capítulo). Por ejemplo, puede especificar:
	<code>>log_type = SYSLOG local0</code>

Tabla 8-3. Variables del archivo de configuración de xinetd (cont.)

Variable	Descripción
	<p>En forma opcional, también puede incluir el nivel de registro. Por ejemplo:</p> <pre>>log_type = SYSLOG local0 info</pre> <p>Si se especifica FILE también debe especificar en cuál nombre de archivo se va hacer el registro. También de manera opcional, puede especificar el límite cambiante en el tamaño del archivo. El límite cambiante en un tamaño de archivo es en donde se generará un mensaje de registro adicional, indicando que el archivo se ha hecho demasiado grande. Si se especifica un límite cambiante, también se debe especificar un límite fijo. En el límite fijo, no se hará registro adicional. Si el límite fijo no se define de manera explícita, se fija como 1% más alto que el cambiante. Un ejemplo de la opción FILE es como se da a continuación:</p> <pre>log_type = FILE /var/log/mylog</pre>
log_on_success	Especifica cuál información se registra en una conexión exitosa. Las opciones incluyen PID para registrar la ID del proceso del servicio que procesó la solicitud, HOST para especificar el anfitrión remoto que se conecta al servicio, USERID para registrar el nombre del usuario remoto (si dispone de él), EXIT para registrar el estado de salida o la señal de terminación del proceso o DURATION para registrar el tiempo que duró la conexión.
port	Especifica el puerto de la red bajo el cual se ejecutará el servicio. Si el servicio se encuentra en la lista de /etc/services , este número de puerto debe ser igual al valor especificado allí.
interface	Permite a un servicio ligarse a una interfaz específica y sólo ser el único del que se dispone allí. El valor es la dirección IP de la interfaz a la que usted desea que este servicio se ligue. Un ejemplo de esto es ligar servicios menos seguros (como Telnet) a una interfaz interna y físicamente segura, en un contrafuegos, y no concederle la interfaz externa y más vulnerable que está fuera de éste.
cps	El primer argumento especifica el número máximo de conexiones por segundo que se permite manejar a este servicio. Si la velocidad sobrepasa este valor, el servicio se desactiva en forma temporal durante el número de segundos del segundo argumento. Por ejemplo:
	<pre>>cps = 10 30</pre> <p>Esto desactivará un servicio durante 30 segundos, si la rapidez de las conexiones sobrepasa la de 10 conexiones por segundo.</p>

Tabla 8-3. Variables del archivo de configuración de xinetd (cont.)

Ejemplos: una entrada sencilla de servicio y activación y desactivación de un servicio

Usando el servicio **finger** como ejemplo, echemos una mirada a una de las entradas más sencillas con **xinetd**:

```
# default: on
# descripción: el servidor finger da respuesta a las solicitudes tipo finger. \
#               Finger es un protocolo que permite a usuarios remotos ver información como \
#               el nombre de acceso y la hora de la última entrada para usuarios locales.
service finger
{
    socket_type      = stream
    wait             = no
    user             = nobody
    server           = /usr/sbin/in.fingerd
}
```

Como puede ver, la entrada se explica por sí misma. El nombre del servicio es **finger** y, debido al **socket_type**, sabemos que éste es un servicio TCP. La variable **wait** nos dice que puede haber múltiples procesos **finger** ejecutándose en forma concurrente. La variable **user** nos dice que “**nobody**” será el propietario del proceso. Por último, el nombre del proceso que se está ejecutando es **/usr/sbin/in.fingerd**.

Con nuestra comprensión de lo que es una entrada **xinetd** de servicio, intentemos activar y desactivar un servicio.

Activación/desactivación del servicio echo

Si quiere un sistema seguro, hay posibilidades de que usted ejecute unos cuantos servicios; ¡hay algunas personas que incluso no ejecutan **xinetd** en lo absoluto! Se requieren unos cuantos pasos para activar o desactivar un servicio. Por ejemplo, para activar un servicio primero activaría ese servicio en el archivo de configuración de **xinetd** (o **inetd.conf** si, por el contrario, está usando **inetd**), reiniciaría el servicio **xinetd** y, por último, probaría las cosas para asegurarse que tiene el comportamiento que espera. Para desactivar un servicio sólo se sigue el procedimiento opuesto.

NOTA El servicio que estaremos examinando es el **echo**. Este servicio es interno a **xinetd**; es decir, no se proporciona para cualquier demonio externo.

Recorramos los pasos de este proceso:

1. Use cualquier editor de textos sencillo con el fin de editar el archivo **/etc/xinetd.d/echo** y cambie la variable **disable** a **no**:

```
# default: off
# descripción: un servicio interno de xinetd en el cual los caracteres de echo
#               regresan a los números de los clientes. Esta es la versión tcp.
service echo
```

```
{
    disable = no
    type      = INTERNAL
    id        = echo-stream
    socket_type = stream
    protocol   = tcp
    user       = root
    wait       = no
}
```

2. Guarde sus cambios hechos al archivo y salga del editor.
3. Reinicie el servicio **xinetd**. En Fedora Core o RHEL, teclee

```
[root@serverA ~]# service xinetd restart
```

Note que para otras distribuciones que no tienen disponible el comando **service**, puede enviar en su lugar una señal HUP a **xinetd**. En primer lugar, encuentre la ID del proceso (PID) de **xinetd**, usando el comando **ps**. Enseguida, use el comando **kill** para enviar la señal HUP a la ID del proceso de **xinetd**. Podemos verificar que el reinicio funcionó mediante el uso del comando **tail** para ver unos cuantos de los últimos mensajes del archivo **//var/log/messages**. Los comandos para hallar la PID de **xinetd**, anular **xinetd** y ver los archivos de registro son

```
[root@serverA ~]# ps -C xinetd
  PID TTY          TIME CMD
31430 ?        00:00:00 xinetd
[root@serverA ~]# kill -1 31430
[root@serverA ~]# tail /var/log/messages
Jun  2 08:49:36 serverA xinetd[31430]: xinetd Version 2009.03.28 started with
libwrap options compiled in.
Jun  2 08:49:36 serverA xinetd[31430]: Started working: 1 available service
Jun  2 08:50:22 serverA xinetd[31430]: Starting reconfiguration
Jun  2 08:50:22 serverA xinetd[31430]: readjusting service echo
Jun  2 08:50:22 serverA xinetd[31430]: Reconfigured: new=0 old=1 dropped=0
```

4. Envíe un mensaje Telnet al puerto (puerto 7) del servicio **echo** y vea si, en verdad, el servicio se está ejecutando. Teclee

```
[root@serverA ~]# telnet localhost 7
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
```

La salida que usted obtenga debe ser semejante a la dada, si se ha activado el servicio **echo**.

Puede teclear cualquier carácter en su teclado, en el mensaje de Telnet, y observar el carácter que **echo** le regresa (repite).

Como puede ver, el servicio **echo** es uno de aquellos servicios terriblemente útiles y salvavidas sin el que los usuarios y los administradores de sistemas *no pueden* desempeñarse.

Este ejercicio lo llevó a través de la capacitación de un servicio, editando directamente su archivo de configuración en **xinetd**. Es un proceso sencillo la activación o la desactivación de un servicio. Pero, en realidad, debe regresar y asegurarse que, en efecto, el servicio está desactivado (si eso es lo que usted quiere), probándolo en realidad. Usted no quiere pensar que ha desactivado Telnet y todavía lo tiene ejecutándose.

SUGERENCIA También puede activar o desactivar un servicio que se ejecuta bajo **xinetd**, mediante el uso de la utilidad **chkconfig**, de la cual se dispone en Fedora Core, RHEL, SuSE y la mayor parte de las otras variedades de Linux. Por ejemplo, para desactivar el servicio **echo**, que desactivó en forma manual, sólo emita el comando **chkconfig echo off**.

EL DEMONIO SYSLOGD

Con tanto que pasa en cualquier momento, en especial con los servicios que están desconectados de una ventana de terminal, es necesario proporcionar un mecanismo estándar mediante el cual se puedan registrar los eventos y mensajes especiales. En Linux, se usa el demonio **syslogd** para dar este servicio.

El demonio **syslogd** proporciona un medio estandarizado de realizar el registro cronológico. En muchos otros sistemas UNIX se emplea un demonio compatible, a fin de dar de este modo un medio para el registro de plataformas cruzadas sobre la red. Esto resulta especialmente valioso en un entorno heterogéneo grande, en donde es necesario centralizar la recolección de entradas de registro con el fin de obtener una imagen exacta de lo que está pasando. Podría igualar este sistema de recursos de registro cronológico con el Windows NT System Logger.

Los archivos de registro, en los que **syslogd** escribe, son archivos de texto directo, que por lo común están almacenados en el directorio **/var/log**. Cada entrada de registro consta de una sola línea que contiene la fecha, la hora, el nombre del anfitrión, el nombre del proceso, la PID y el mensaje de ese proceso. Una función de uno a otro lado del sistema que se encuentra en la biblioteca C estándar suministra un mecanismo fácil para generar los mensajes de registro. Si el lector no se siente a gusto con escribir un código, sino que quiere generar entradas en los registros, tiene la opción de usar el comando **logger**.

Como puede imaginar, una herramienta con la importancia de **syslogd** es algo que se arranca como parte de los scripts de inicialización. Todas las distribuciones de Linux que usted usaría en un entorno de servidores ya harán esto en lugar de usted.

Invocación de **syslogd**

Si en realidad necesita iniciar **syslogd** en forma manual o de modificar el script que lo arranca en la inicialización, necesitará estar al tanto de los parámetros de la línea de comandos de **syslogd**, los cuales se muestran en la tabla 8-4.

El archivo **/etc/syslog.conf**

El archivo **/etc/syslog.conf** contiene la información de configuración que **syslogd** necesita para ejecutarse. El formato del archivo es un poco desacostumbrado, pero es probable que sea suficiente la configuración predeterminada que usted tenga, a menos que necesite enviar información determinada en archivos específicos o, quizás, quiera enviar mensajes locales de registro hacia máquinas remotas de registro que puedan aceptarlos.

Parámetro	Descripción
-d	Modo de corrección de errores. Normalmente, en el arranque, syslogd se separa de la terminal actual y empieza a ejecutarse en segundo plano. Con la opción -d , syslogd retiene el control de la terminal e imprime la información relativa a la corrección de errores conforme se registran los mensajes. Es muy improbable que necesite esta opción.
-f config	Especifica un archivo de configuración como una alternativa por el <code>/etc/syslog.conf</code> predeterminado.
-h	De modo predeterminado, syslogd no hace que sigan su camino mensajes que le envían y que en realidad estuvieran destinados para otro anfitrión. Precaución: Si usa este parámetro, corre el riesgo de ser usado como parte de un ataque de negación de servicio.
-l hostlist	Esta opción le permite hacer la lista de los anfitriones para los cuales está deseando realizar registro cronológico. Cada nombre de anfitrión debe ser su nombre sencillo, no su nombre completamente calificado de dominio (FQDN, <i>fully qualified domain name</i>). Puede hacer listas de múltiples anfitriones, siempre y cuando se separen por medio de dos puntos; por ejemplo,
	-l toybox:serverB
-m interval	De modo predeterminado, syslogd genera una entrada de registro cada 20 minutos como un mensaje de "sólo para que usted sepa que me estoy ejecutando". Esto es para los sistemas que pueden no estar ocupados (si está observando el registro cronológico del sistema y no ve un solo mensaje en más de 20 minutos, sabrá de hecho que algo anda mal). Al especificar un valor numérico para interval , puede indicar el número de minutos que syslogd debe esperar antes de generar otro mensaje.
-r	De modo predeterminado, como una precaución de seguridad, el demonio syslogd rechaza los mensajes que le envían desde la red. Este parámetro de la línea de comandos activa esta característica.
-s domainlist	Si está recibiendo entradas de syslogd que muestran el FQDN completo, puede hacer que el propio syslogd arranque el nombre del dominio y sólo deje el nombre del anfitrión. Sencillamente haga una lista de los nombres de dominio por separados por dos puntos, como el parámetro para la opción -s . Por ejemplo:
	-s conspiracy.com:wealthy.com

Tabla 8-4. Parámetros de la línea de comandos de **syslogd**

Clasificaciones de los mensajes del registro cronológico

Antes de que pueda comprender el propio formato del archivo `/etc/syslog.conf`, tiene que entender cómo se clasifican los mensajes del registro cronológico. Cada mensaje tiene un *recurso* y una *prioridad*. El recurso le dice desde cuál subsistema se originó el mensaje y la prioridad le informa acerca de cuán importante es ese mensaje. Estos dos valores están separados por un punto.

Los dos valores tienen equivalentes en cadena, lo que los hace más fáciles de recordar. En las tablas 8-5 y 8-6, respectivamente, se da una lista de los equivalentes en cadena para el recurso y la prioridad.

NOTA Los niveles de prioridad se encuentran en el orden de lo estricto que son según **syslogd**. De este modo, **debug** no se considera estricto en lo absoluto y **emerg** es el más crucial. Por ejemplo, la cadena de la combinación recurso y prioridad **mail.crit** indica que hay un error crítico en el subsistema de correo (por ejemplo, se ha agotado el espacio de disco). **syslogd** considera este mensaje más importante que **mail.info**, que sencillamente hace notar la llegada de otro mensaje.

Equivalente en cadena del recurso	Descripción
auth	Mensajes de autenticación
authpriv	Esencialmente lo mismo que auth
cron	Mensajes generados por el subsistema cron
daemon	Clasificación genérica para los demonios de servicios
kern	Mensajes del núcleo
Lpr	Mensaje del subsistema de impresoras
Mail	Mensajes del subsistema de correo (incluyendo registros de correo)
Mark	Obsoleta, pero puede ser que la encuentre en algunos libros que la discutan; syslogd sencillamente la ignora
News	Mensajes a través del subsistema NNTP
security	Lo mismo que auth ; no debe usarse
syslog	Mensajes internos del propio syslog
User	Mensajes genéricos de los programas de usuarios
Uucp	Mensajes del subsistema UUCP (UNIX hacia copia de UNIX)
Local0-local9	Niveles genéricos de los recursos cuya importancia se puede decidir con base en las necesidades de usted

Tabla 8-5. Los equivalentes en cadena para el recurso en `/etc/syslog.conf`

Equivalente en cadena del recurso	Descripción
<code>debug</code>	Informes de corrección de errores
<code>info</code>	Información diversa
<code>notice</code>	Informes importantes, pero no necesariamente malas noticias
<code>warning</code>	Situación potencialmente peligrosa
<code>warn</code>	Lo mismo que <code>warning</code> ; no debe usarse
<code>err</code>	Una condición de error
<code>error</code>	Lo mismo que <code>err</code> ; no debe usarse
<code>crit</code>	Situación crítica
<code>alert</code>	Un mensaje que indica un suceso importante
<code>emerg</code>	Una situación de emergencia

Tabla 8-6. Equivalentes en cadena para los niveles de prioridad en /etc/syslog.conf

Además de los niveles de prioridad de la tabla 8-6, **syslogd** también interpreta los comodines. Por consiguiente, el lector puede definir una clase completa de mensajes; por ejemplo, **mail**.* se refiere a todos los mensajes relacionados con el subsistema de correo.

Formato de /etc/syslog.conf

A continuación se tiene el formato de cada línea en el archivo de configuración:

facility/priority combinations separated by semicolons file/process/host to log to

Por ejemplo:

kern.info; kern.err /var/log/kernel-info

file/process/host to log to

La ubicación a la cual **syslogd** puede enviar los mensajes de registro también es bastante flexible. Puede guardar mensajes en los archivos y enviar mensajes a los FIFO (First in-First out, primero dentro-primero fuera), a una lista de usuarios o (en el caso de registro cronológico centralizado para un sitio grande) a un anfitrión de registro maestro. Con el fin de diferenciar estos elementos de ubicación, se aplican las reglas siguientes a la entrada de ubicación.

- ▼ Si la ubicación principia con una barra diagonal (/), el mensaje está yendo hacia un archivo.
 - Si la ubicación principia con un tubo (|), el mensaje está yendo hacia una tubería nombrada (FIFO).
 - ▲ Si la ubicación principia con una @, el mensaje está yendo hacia un anfitrión.

En la tabla 8-7, se muestran ejemplos de entradas de ubicación.

Estilo de ubicación	Descripción
Regular file (e.g., /var/log/messages)	Un archivo. Note que si pone como prefijo una barra diagonal al nombre del archivo, syslogd no sincronizará el sistema de archivos después de la escritura. Esto significa que usted corre el riesgo de perder algunos datos si se tiene una caída antes de que el sistema tenga posibilidad de limpiar sus memorias intermedias. Por otra parte, si una aplicación está siendo demasiado prolífica acerca de su registro cronológico, usted ganará rendimiento con el uso de esta opción. Recuerde: si quiere enviar mensajes a la consola, necesita especificar /dev/console.
!/tmp/mypipe	Una tubería nombrada. Este tipo de archivo se crea con el comando mknod command. Con syslogd alimentando desde un extremo de la tubería, puede tener otro programa ejecutándose que lea en el otro extremo de la tubería. Ésta es una manera eficaz de tener programas analizando la salida de registro, buscando situaciones críticas, de modo que se pueda avisar a usted, si es necesario.
@loghost	Un nombre de anfitrión. Este ejemplo enviará el mensaje a loghost . Entonces el demonio syslogd en loghost registrará el mensaje.

Tabla 8-7. Ejemplos de entradas de ubicación

Si no introduce un carácter especial antes de la entrada de ubicación, **syslogd** supone que la ubicación es una lista de usuarios separados por comas, quienes tendrán el mensaje escrito en sus pantallas. Si usa un asterisco (*), **syslogd** enviará el mensaje a todos los usuarios quienes hayan tenido acceso.

Como es costumbre, cualquier línea que empieza con un símbolo de número (#) es un comentario. Ahora echemos una mirada a algunos ejemplos de entradas del archivo de configuración:

```
# Regístrense todos los mensajes de correo en un lugar.
mail.* /var/log/maillog
```

Este ejemplo hace ver que todas las prioridades en el recurso de correo deben tener sus mensajes colocados en el archivo **/var/log/maillog**.

Considere el ejemplo siguiente:

```
# Enviar mensajes de emergencia a los usuarios que aparecen en la lista, más registrar
# esos mensajes en el sistema remoto.
*.emerg      @loghost,yyang,root,dude
```

En este ejemplo, el lector ve que cualquier recurso con un nivel de registro de **emerg** se envía hacia otro sistema que ejecuta **syslogd**, llamado **loghost**. Asimismo, si los usuarios yyang, root o du-de están conectados, el mensaje que se está registrando se escribe en la consola de los mismos.

También puede especificar selectores múltiples en una sola línea para un solo evento. Por ejemplo:

```
*.info;mail.none;authpriv.none          /var/log/messages
```

Archivo muestra de /etc/syslog.conf

El siguiente es un archivo **syslog.conf** completo:

```
# Registrar todos los mensajes del núcleo en la consola.
# Registrar mucho más obstruye la pantalla.
#kern.*                                     /dev/console
# Registrar todo (excepto correo) del nivel info o superior.
# ¡No registrar mensajes privados de autenticación!

*.info;mail.none;authpriv.none;cron.none      /var/log/messages
# El archivo authpriv tiene acceso restringido.
authpriv.*                                    /var/log/secure

# Registrar todos los mensajes de correo en un lugar.
mail.*                                         - /var/log/maillog

# Registrar el material de cron
cron.*                                         /var/log/cron

# Enviar mensajes de emergencia a todos
*.emerg                                         *

# Guardar los errores de noticias de nivel crit y superior en un archivo especial.
uucp,news.crit                                  /var/log/spooler

# Guardar también los mensajes de inicialización en boot.log
local7.*                                         /var/log/boot.log
```

CRON

El programa **cron** permite a cualquier usuario del sistema hacer que un programa se ejecute en cualquier fecha, a cualquier hora o en un día particular de la semana, hasta el minuto. Usar **cron** es una manera en extremo eficiente de automatizar su sistema, generar informes de manera regular y realizar otras tareas periódicas. (Usos no tan honestos de **cron** incluyen: ¡hacer que se ejecute un sistema que le envíe un mensaje a su buscador cuando quiere salirse de una reunión!)

Como los otros servicios que hemos discutido en este capítulo, **cron** lo arrancan los scripts de inicialización y, lo más probable, es que ya esté configurado para usted. Una revisión rápida de la lista de procesos debe mostrarlo ejecutándose silenciosamente en segundo plano:

```
[root@serverA ~]# ps aux | grep crond | grep -v grep
root      2422  0.0  0.3  2256  764 ?          Ss   Mar17  0:25 crond
```

El servicio **cron** funciona acercándose una vez cada minuto y verificando el archivo **crontab** de cada usuario. Este archivo contiene la lista de eventos de los usuarios que desean que se ejecuten en una fecha y hora particulares. Cualesquiera eventos que coincidan con la fecha y hora en curso se ejecutan.

El propio comando **crond** no requiere parámetros de la línea de comandos o señales especiales para indicar un cambio en el estado.

El archivo crontab

La herramienta que permite editar las entradas que deben ser ejecutadas por **crond** es **crontab**. En esencia, todo lo que hace es verificar el permiso de usted para modificar sus ajustes en **cron** y, a continuación, llama a un editor de textos de modo que usted pueda hacer sus cambios. Una vez que usted ha terminado, **crontab** coloca el archivo en la ubicación correcta y le contesta con un mensaje.

crontab determina si usted tiene o no el permiso apropiado al revisar el archivo **/etc/cron.allow** y, a continuación, el **/etc/cron.deny**. Si existe cualquiera de estos dos archivos, usted debe encontrarse explícitamente en una lista allí para que sus acciones se efectúen. Por ejemplo, si existe el archivo **/etc/cron.allow**, su nombre de usuario debe encontrarse en la lista de ese archivo para que usted pueda editar sus entradas **cron**. Por otra parte, si el único archivo que existe es **/etc/cron.deny**, a menos que su nombre de usuario se encuentre en lista allí, implícitamente tiene permiso para editar sus ajustes de **cron**.

El archivo en el que se encuentra la lista de sus tareas **cron** (a menudo conocido como el archivo **crontab**) se formatea como se indica enseguida. Todos los valores deben aparecer en la lista como enteros.

Minute Hour Day Month DayOfWeek Command

Si quiere tener entradas múltiples para una columna en particular (por ejemplo, quiere que un programa se ejecute a las 4:00 A.M., las 12:00 P.M. y las 5:00 P.M.), lo que necesita entonces es tener cada uno de estos valores de tiempo en una lista separados por comas. Asegúrese que no haya ningún tipo de espacio en la lista. Para ejecutar el programa a las 4:00 A.M., 12:00 P.M. y 5:00 P.M., la lista de valores **Hour** quedaría **4,12,17**. Versiones más recientes de **cron** le permiten usar una notación más corta para suministrar los campos. Por ejemplo, si quiere ejecutar un proceso cada dos minutos, sólo necesita poner **/2** como la primera entrada. Advierta que en **cron** se usa el formato militar de tiempo.

Para la entrada **DayOfWeek**, 0 representa el domingo, 1 representa el lunes, y así sucesivamente hasta 6 que representa el sábado.

Cualquier entrada que tiene un comodín de un solo asterisco (*) se ajustará a cualquier minuto, hora, día, mes o día de la semana, cuando se use en la columna correspondiente.

Cuando las fechas y horas del archivo se ajustan a la fecha y hora en curso, el comando se ejecuta como el usuario que fijó el **crontab**. Para cualquier salida generada se envía un correo de

regreso al usuario. Es obvio que esto puede dar como resultado un buzón lleno de mensajes, de modo que es importante ser ahorrativo con sus informes. Una buena manera de mantener un control sobre el volumen es hacer salir sólo condiciones de error y hacer que cualquier salida inevitable se envíe a `/dev/null`.

Veamos algunos ejemplos. La entrada que sigue ejecuta el programa `/bin/ping -c 5 serverB` cada cuatro horas:

```
0 0,4,8,12,16,20 * * * /bin/ping -c 5 serverB
```

o, usando el método abreviado:

```
0 */4 * * * /bin/ping -c 5 serverB
```

He aquí una entrada que ejecuta el programa `/usr/local/scripts/backup_level_0` a las 10:00 P.M. cada viernes en la noche:

```
0 22 * * 5 /usr/local/scripts/backup_level_0
```

Y, por último, enseguida se da un script para enviar un correo a las 4:01 A.M. el 1 de abril (cualquier día que pueda ser):

```
1 4 1 4 * /bin/mail dad@domain.com < /home/yyang/joke
```

NOTA Cuando `cron` ejecuta los comandos, lo hace con el shell `sh`. Por tanto, cualesquiera variables de entorno que podrían ser usadas por usted, puede ser que no funcionen con `cron`.

Edición del archivo crontab

Editar o crear un trabajo `cron` es tan fácil como editar un archivo normal de texto. Pero debe estar consciente del hecho de que, de modo predeterminado, el programa usará un editor especificado por la variable de entorno EDITOR o VISUAL. En la mayor parte de los sistemas Linux, el editor predeterminado suele ser `vi`. Pero el lector siempre puede cambiar este editor predeterminado por cualquier otro con el que se siente cómodo mediante el ajuste de la variable de entorno EDITOR o VISUAL.

Ahora que ya conoce el formato del archivo de configuración `crontab`, necesita editarlo. No hace esto editando el archivo en forma directa; use el comando `crontab` para editar su archivo `crontab`:

```
[yyang@serverA ~]$ crontab -e
```

Para obtener una lista de lo que está en su archivo `crontab` actual, sólo dé a `crontab` el argumento `-l` para presentar el contenido. Teclee

```
[yyang@serverA ~]$ crontab -l  
no crontab for yyang
```

Según esta salida, el usuario yyang nada tiene en la actualidad en el archivo `crontab`.

RESUMEN

En este capítulo, discutimos algunos servicios importantes del sistema que vienen con la mayor parte de los sistemas Linux. Estos servicios no requieren soporte de red y pueden variar de anfitrión a anfitrión, haciéndolos muy útiles, pues pueden funcionar ya sea que el sistema esté o no en un modo de usuarios múltiples.

Un resumen rápido del capítulo:

- ▼ **init** es madre de todos los procesos que se encuentran en el sistema, con una PID de 1. También controla los niveles de ejecución y se puede configurar a través del archivo `/etc/inittab`.
- **inetd**, aunque ya no se usa más que escasas veces, es el superservidor original que escucha las solicitudes de los servidores en nombre de un gran número de servicios más pequeños que se usan con menos frecuencia. Cuando acepta una solicitud de uno de esos servicios, **inetd** inicia el servicio real y de manera silenciosa dirige los datos entre la red y ese servicio. Su archivo de configuración es `/etc/inetd.conf`.
- **xinetd** es la versión “nueva” del superservidor **inetd** clásico que ofrece más opciones de configuración y mejor seguridad integrada. Su archivo principal de configuración es `/etc/xinetd.conf`.
- **syslog** es el demonio de registro cronológico para todo el sistema. Junto con las entradas de registro generadas por el sistema, **syslog** puede aceptar mensajes de registro sobre la red (siempre que usted active esa característica). Su archivo de configuración es `/etc/syslog.conf`.
- ▲ Por último, el servicio **cron** le permite programar que puedan llevarse a cabo eventos en ciertas fechas y horas, lo cual es formidable para los eventos periódicos como respaldos y recordatorios de correos. Todos los archivos de configuración sobre los cuales se apoya se manejan a través del programa **crontab**.

En cada sección de este capítulo, discutimos cómo configurar un servicio diferente e incluso sugerimos algunos más allá de los ajustes predeterminados que vienen con el sistema. Se recomienda que se asome alrededor de estos servicios y se familiarice con lo que puede realizar con ellos. En torno de estos servicios básicos, se han estructurado muchas herramientas poderosas de automatización, de reunión de datos y de análisis; así como muchas cosas maravillosamente absurdas e inútiles. ¡No tenga miedo de divertirse con ello!

CAPÍTULO 9



Compilación
del núcleo de Linux

Una de las fuerzas más grandes de Linux es que su código fuente se encuentra disponible para quienquiera que lo solicite. ¡La GNU GPL (General Public License, Licencia pública general), bajo la cual se distribuye Linux, incluso le permite arreglar como quiera el código fuente y distribuir sus cambios! Los cambios reales al código fuente (al menos aquellos que deben tomarse con seriedad) pasan por el proceso de unirse al árbol oficial del núcleo. Esto exige extensas pruebas y demostraciones respecto a que los cambios beneficiarán a Linux como un todo. En el justo término del proceso de aprobación, el código recibe un sí o un no final por parte de un grupo central de los desarrolladores originales del proyecto Linux. Éste extenso proceso de aprobación es el que preserva la calidad del código de Linux tan digno de mención.

Para los administradores de sistemas que han usado otros sistemas operativos patentados, este procedimiento de control del código se separa de manera significativa de la filosofía de esperar que la compañía ponga en circulación un parche, un paquete de servicios o algún tipo de "solución caliente". En lugar de tener que avanzar con dificultad a través de relaciones públicas, ingenieros de ventas y otras unidades frontales, tiene la posibilidad de ponerse directamente en contacto con el autor del subsistema y explicarle su problema. Se puede crear un parche y enviárselo, antes de la siguiente publicación oficial del núcleo, y mantenerlo a usted en pie y funcionando.

Por supuesto, la pequeña dificultad de esta disposición del trabajo es que usted necesita ser capaz de compilar por sí mismo un núcleo, en lugar de depender de alguien más para que le suministre un código precompilado. Y, por supuesto, no tendrá que hacer esto con frecuencia, porque los entornos de producción, una vez estables, rara vez necesitan una compilación del núcleo. Pero, si es necesario, usted debe saber qué hacer. Por fortuna, no es difícil.

En este capítulo, pasaremos por el proceso de adquirir un árbol fuente del núcleo, configurarlo, compilarlo y, por último, instalar el resultado final.

PRECAUCIÓN El núcleo es lo primero que carga cuando se inicializa un sistema Linux (por supuesto, ¡después del cargador de inicialización!). Si el núcleo no funciona correctamente, es improbable que se inicialice el resto del sistema. Asegúrese de tener a mano un medio de inicialización de emergencia o rescate, en caso que necesite volver a una antigua configuración (vea la sección sobre GRUB en el capítulo 6).

¿QUÉ ES EXACTAMENTE UN NÚCLEO?

Antes de saltar hacia el proceso de compilación, demos un paso atrás y asegurémonos que usted tiene claro el concepto de lo que es un núcleo y del papel que desempeña en el sistema. Lo más frecuente es que, cuando la gente dice "Linux" suele referirse a una "distribución de Linux". Como se discutió en el capítulo 1, una distribución comprende todo lo necesario para hacer que Linux exista como sistema operativo funcional (por ejemplo, SuSE Linux es un tipo de distribución de Linux). En las distribuciones se usa el código proveniente de varios proyectos de fuente abierta que son independientes de Linux; de hecho, muchos de los paquetes de software mantenidos por estos proyectos también se usan en forma extensiva en otras plataformas semejantes a UNIX. Por ejemplo, el GNU C Compiler, el cual viene con la mayor parte de las distribuciones de Linux, también existe en muchos otros sistemas operativos (probablemente en más sistemas que los que la mayor parte de la gente se da cuenta que existen).

En tales términos, entonces ¿qué *integra* la definición pura de Linux? El *núcleo*. El núcleo de un sistema operativo es el centro de todo el software del sistema. Lo único más fundamental que el núcleo es el propio hardware.

El núcleo tiene muchos trabajos. La esencia de su trabajo es abstraer el hardware subyacente del software y proporcionar un entorno de ejecución para el software de aplicación, a través de llamadas del sistema. En específico, el entorno debe manejar aspectos como la operación en red, el acceso al disco, la memoria virtual y las tareas múltiples; ¡una lista completa de estas tareas abarcaría un capítulo completo por sí misma! El núcleo de hoy de Linux (versión 2.6.*¹) contiene casi seis millones de líneas de código (incluyendo controladores de dispositivos). Como comparación, la sexta edición de UNIX de los Bell Labs, en 1976, tenía aproximadamente 9000 líneas. En la figura 9-1, se ilustra la posición del núcleo en un sistema completo.

Aunque el núcleo es una parte pequeña de un sistema Linux completo, con mucho es el elemento más crítico. Si el núcleo falla o se cae, el resto del sistema se va con él. Por fortuna, Linux puede presumir de la estabilidad de su núcleo. Los *tiempos útiles* (el tiempo que transcurre entre reinicios) para los sistemas Linux con frecuencia se expresan en años. En efecto, los sistemas UNIX, en general, se atribuyen tiempos útiles significativamente largos.

MANERA DE HALLAR EL CÓDIGO FUENTE DEL NÚCLEO

Es probable que su distribución de Linux tenga el código fuente para la versión (o versiones) específica(s) del núcleo que la soporta, que se encuentra disponible en una forma u otra. Éstas podrían estar en la forma de un binario compilado (*.src.rpm), un rpm fuente (*.srpm) o algo por el estilo.

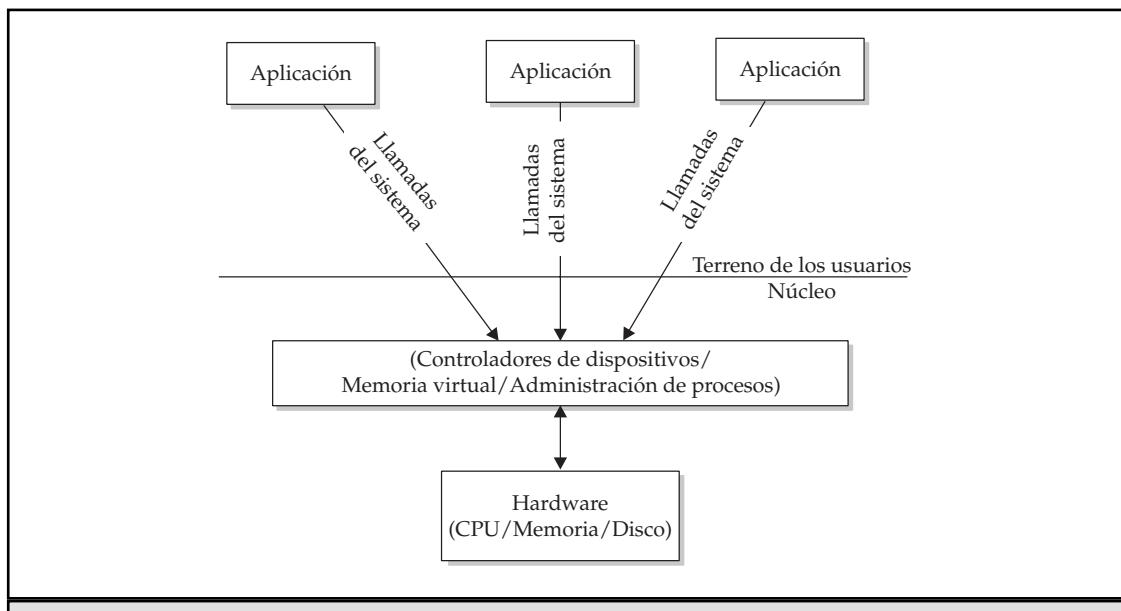


Figura 9-1. Representación visual de la manera en que el núcleo de Linux se acomoda en un sistema completo

Si tiene necesidad de descargar una versión diferente (posiblemente más reciente) que la que le proporciona su distribución particular de Linux, el primer lugar para buscar el código fuente es en el sitio Web oficial del núcleo: <http://www.kernel.org/>. El árbol de kernel.org representa el “Árbol de Linus”. En este sitio se mantiene una lista de los sitios Web que sirven de espejos para la fuente del núcleo, así como de toneladas de otro software de fuente abierta y de utilidades para fines generales.

Debajo del vínculo del sitio para descargar el núcleo, encontrará una lista de sitios Web espejo basada en los códigos de los países. Aun cuando se puede conectar a cualquiera de ellos, es probable que obtenga el mejor rendimiento si se adhiere a su propio país. Vaya a <http://www.xx.kernel.org/>, en donde xx es el código de país de Internet que corresponda a donde usted vive. Para Estados Unidos, esta dirección es <http://www.us.kernel.org/>.

NOTA Hay una tendencia común en la gente de querer descargar la fuente más prístina posible, ¡y de dónde más se obtendría la fuente más prístina posible sino del propio kernel.org! Pero existen varias otras razones que es probable sean buenas para obtener las fuentes de otros lugares que no sean kernel.org. Por una parte, siempre puede verificar la firma o md5sum del archivo que descarga con la firma oficial. La mayor parte de los sitios que sirven de espejo para kernel.org sincronizan en forma regular, y con frecuencia, sus sitios con el de éste. Con todo esto presente, es probable que esté bien que usted descargue el árbol fuente de un espejo que geográficamente le quede más cercano.

Obtención de la versión correcta del núcleo

El sitio Web que tiene la lista de los núcleos de los que dispone contendrá carpetas para v1.0, v1.1, y así sucesivamente, y para v2.5, v2.6, etcétera. Antes de que siga su inclinación natural de obtener la versión más reciente, asegúrese de que entiende la manera en que funciona el sistema de establecer las versiones de Linux.

Debido a que el modelo de desarrollo de Linux alienta las colaboraciones del público, la versión más reciente del núcleo debe estar accesible a todos, en cualquier momento. Sin embargo, esto presenta un problema: el software que está pasando por actualizaciones significativas puede ser inestable y no de calidad de producción.

Para dar vuelta a este problema, los primeros desarrolladores de Linux adoptaron un sistema de usar núcleos con número impar (1.1, 1.3, 2.1, 2.3, etc.) para indicar un ciclo de diseño y desarrollo. Por tanto, los núcleos con números impares llevan la renuncia de que pueden no ser estables y no se deben usar para situaciones para las cuales la confiabilidad es un deber. Por lo general, estos núcleos en desarrollo se publican con una velocidad muy alta, ya que existe tanta actividad en torno a ellos, ¡se pueden publicar nuevas versiones de núcleos en desarrollo con tanta frecuencia como dos veces por semana!

Por otra parte, los núcleos con número par (1.0, 1.2, 2.0, 2.2, 2.4, 2.6, etc.) se consideran sistemas listos para la producción. Se les ha dejado madurar en el uso (y escrutinio) público. A diferencia de los núcleos en desarrollo, los de producción se publican con una velocidad mucho más lenta y contienen sobre todo arreglos de errores.

La versión del núcleo con la que vamos a trabajar en la sección que sigue es 2.6.12.1, la cual se puede obtener en <http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.12.1.tar.gz>.

SUGERENCIA Puede usar la utilidad `wget` para descargar con rapidez la fuente del núcleo hacia su directorio actual de trabajo, al teclear

```
# wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-  
2.6.12.1.tar.gz
```

Desempaque el código fuente del núcleo

La mayor parte de los paquetes con los que el lector ha trabajado hasta ahora probablemente han sido paquetes RPM y existe una gran posibilidad de que esté acostumbrado a usar las herramientas que vienen con el sistema (como RPM o Yast) para administrar los paquetes. El código fuente del núcleo es un poco diferente y requiere algo de participación del usuario. Recorramos los pasos para desempacar el núcleo.

La fuente del núcleo consta de un montón de archivos diferentes y, debido al número y tamaño verdaderos de estos archivos en forma colectiva, resulta útil comprimirlos y ponerlos todos en una sola estructura de directorio. La fuente del núcleo que descargará de Internet es un archivo que ha sido comprimido y se le ha aplicado la utilidad tar. Por lo tanto, para usar la fuente, descomprima el archivo fuente y desaplique tar en él. Esto es lo que significa desempacar el núcleo. En general, en realidad es un proceso directo.

La ubicación tradicional para el árbol fuente del núcleo en el sistema local de archivos es el directorio `/usr/src`. En el resto de este capítulo, supondremos que usted está trabajando fuera del directorio `/usr/src`.

NOTA A veces, algunas distribuciones de Linux tienen un vínculo simbólico en el directorio `/usr/src`. Este vínculo suele nombrarse "linux" y, por lo común, es un vínculo hacia un árbol predeterminado o hacia el árbol fuente más reciente del núcleo. ¡Algunos paquetes de software de terceras partes se apoyan en este vínculo para compilar y estructurar con propiedad!

Copie la tarball (bola de brecha) del núcleo que descargó con anterioridad en el directorio `/usr/src`. Use el comando `tar` para desempacar y descomprimir el archivo. Teclee

```
[root@serverA src]# tar xvzf linux-2.6.12.1.tar.gz
```

Oirá zumbar su disco duro durante unos cuantos instantes, conforme se ejecuta el comando; ¡después de todo, la fuente del núcleo es un archivo grande!

SUGERENCIA Tómese un momento para revisar lo que está dentro del árbol fuente del núcleo. Por lo menos, tendrá una posibilidad de ver qué clase de documentación se embarca con un núcleo en existencia. Una buena parte de la documentación del núcleo está almacenada de manera conveniente en el directorio `Documentation`, al final del árbol fuente del propio núcleo.

ESTRUCTURACIÓN DEL NÚCLEO

De modo que ahora tiene un árbol desempacado del núcleo sólo esperando que se le estructure. En esta sección. Vamos a revisar el proceso de configurar y estructurar el núcleo. Esto contrasta con los sistemas operativos como Windows 2000, los cuales vienen preconfigurados y, por consiguiente, contienen soporte para muchas características que puede usted querer o no.

La filosofía de diseño de Linux permite al individuo decidir acerca de las partes importantes del núcleo. (Por ejemplo, si no tiene un subsistema SCSI, ¿qué caso tiene desperdiciar memoria para soportarlo?) Este diseño individualizado tiene el importante beneficio de permitirle adelgazar la lista de características, de modo que Linux pueda ejecutarse de manera tan eficiente como sea posible. Ésta también es una de las razones por las que es posible ejecutar Linux en varias estructuras de hardware, desde sistemas de baja finalidad, pasando por sistemas incrustados, hasta sistemas realmente de alta finalidad. Puede encontrar que una caja incapaz de soportar un servidor basado en Windows es más que capaz de soportar un OS basado en Linux.

Se requieren dos pasos en la estructuración de un núcleo: configurar y compilar. En este capítulo, no queremos llegar a los aspectos específicos de la configuración, lo cual sería difícil debido a la evolución a pasos rápidos del núcleo de Linux. Sin embargo, una vez que comprenda el proceso básico, debe ser capaz de aplicarlo de una versión a otra. En beneficio de la discusión. Citaremos ejemplos del núcleo v2.6.* que desempacamos en la sección anterior.

El primer paso en la estructuración del núcleo es configurar sus características. Por lo común, la lista de características deseadas del lector se basará en cualquiera que sea el hardware que necesita soportar. Por supuesto, esto significa que necesitará una lista de ese hardware.

En un sistema que ya está ejecutando Linux, el comando que sigue proporcionará una lista de todo el hardware conectado al sistema a través del bus PCI:

```
[root@serverA ~]# lspci
```

Con esta lista del hardware, está listo para configurar el núcleo.

Evite mejoras innecesarias

Tenga presente que, si tiene un sistema en funcionamiento que es estable y que se comporta bien, existen pocas razones para mejorar el núcleo, a menos que tenga usted una de estas condiciones

- ▼ Existe una solución de seguridad que debe aplicar.
- Existe una característica específica nueva en una emisión estable que necesita.
- ▲ Existe una solución específica para errores que le impacta.

En el caso de una solución de seguridad, decida si, en realidad, el riesgo le impacta: por ejemplo, si se encuentra el aspecto de seguridad en un controlador de dispositivo que no usa, entonces no existe razón para la modernización. En el caso de la emisión de la solución de un error, lea con cuidado las notas de la emisión y decida si las soluciones en realidad le impactan; si tiene un sistema estable, mejorar el núcleo con parches que nunca use sólo es introducir riesgo sin razón. En los sistemas de producción, el núcleo no debe sencillamente mejorarse sólo para tener “el núcleo más reciente”; debe haber una razón en verdad poderosa para la mejora.

Preparación para configurar el núcleo

Ahora que tenemos una idea aproximada de los tipos de hardware y de las características que nuestro nuevo núcleo necesita para soportar, podemos empezar la configuración real. Pero, en primer lugar, vaya alguna información básica:

El árbol fuente del núcleo de Linux contiene varios archivos llamados **Makefile** (un makefile es tan sólo un archivo de texto que describe las relaciones entre los archivos en un programa). Estos makefiles ayudan a pegar los miles de otros archivos que constituyen en conjunto la fuente del núcleo. Lo que es más importante para nosotros en este punto: los makefiles también contienen objetivos. Los objetivos son los comandos o directivas que ejecuta el programa **make**.

El **Makefile** en la raíz del árbol fuente del núcleo contiene objetivos específicos que se pueden usar en la preparación del entorno de la estructuración de ese núcleo, la configuración de éste, la compilación del mismo, su instalación, etcétera. Enseguida, se discuten con más detalle algunos de los objetivos:

- ▼ **make mrproper** Este objetivo limpia el entorno de la estructura de cualesquiera archivos viciados y dependencias que podrían haber quedado de una estructura previa del núcleo. Se limpiarán (borrarán) todas las configuraciones anteriores del núcleo del entorno de la estructura.
- **make clean** Este objetivo no realiza un trabajo tan completo como el “mrproper”. Sólo borra la mayor parte de los archivos generados. No borra el archivo de configuración del núcleo (**.config**).
- **make menuconfig** Este objetivo llama a una interfaz de editor basada en texto con menús, listas de radio y cuadros de diálogo basados en texto para configurar el núcleo.
- **make xconfig** Ésta es una herramienta de configuración del núcleo basada en X Window que se apoya en las bibliotecas Qt de desarrollo gráfico. Estas bibliotecas se usan en las aplicaciones basadas en KDE.
- **make gconfig** Este objetivo también llama a una herramienta de configuración del núcleo basada en X Window. Pero se apoya en el juego de herramientas GTK2 (GIMP). Este juego de herramientas GTK2 se usa mucho en el mundo del escritorio de GNOME.
- ▲ **make help** Este objetivo le mostrará todos los demás objetivos posibles de make y también sirve como una fuente del rápido sistema de ayuda en línea.

Para configurar el núcleo en esta sección, sólo usaremos uno de los objetivos. En particular, usaremos el comando **make xconfig**. El editor de configuración del núcleo de **xconfig** es una de las herramientas más populares para configurar los núcleos de la serie 2.6 de Linux. El editor gráfico tiene una interfaz sencilla y limpia, y su uso es *casi* intuitivo.

Pero antes de empezar la configuración real del núcleo, debe limpiar (preparar) el entorno de la estructura del núcleo mediante el uso del comando **make mrproper**. Teclee

```
[root@serverA linux-2.6.12.1] # make mrproper
```

Configuración del núcleo

A continuación, recorreremos por pasos el proceso de configuración de un núcleo de la serie 2.6.* de Linux. Para examinar algunas de las entrañas de este proceso, activaremos el soporte de una característica muy específica que pretendemos debe ser soportada sobre el sistema. Una vez que comprenda cómo funciona esto, puede aplicar el mismo procedimiento a fin de agregar soporte para cualquier otra característica nueva del núcleo que quiera. Específicamente, activaremos el soporte para el sistema de archivos NTFS en nuestro núcleo personal.

Las distribuciones más modernas de Linux que se embarcan con los núcleos de la serie 2.6.* (en donde el asterisco es un comodín que representa el número completo de la versión del núcleo) también cuentan con un archivo de configuración del núcleo para poner en ejecución el núcleo del que se disponga en el sistema local de archivos, como un archivo comprimido o normal. En nuestro sistema muestra, en el que se ejecuta el Fedora Core Linux, este archivo reside en el directorio **/boot** y suele nombrarse en alguna forma como "config-2.6.*". El archivo de configuración contiene una lista de las opciones que se activaron para el núcleo particular que representa. Un archivo de configuración semejante a éste es el que pretendemos crear en el curso del proceso de configuración del núcleo. La única diferencia entre el archivo que crearemos y el ya hecho es que hemos agregado nuestra personalización adicional.

El uso de un archivo de configuración preexistente y conocido como un armazón para la creación de nuestro propio archivo personalizado nos ayuda a garantizar que no desperdiciaremos demasiado tiempo duplicando el esfuerzo que otros ya han realizado ¡para hallar lo que funciona y lo que no funciona!

Los pasos que siguen cubrirán la manera de compilar el núcleo, después de que, en primer lugar, haya recorrido la configuración de ese núcleo. Estaremos usando la utilidad de configuración Graphical Kernel, de modo que su X Window System necesita estar levantado y ejecutándose.

1. Para empezar, copiaremos y daremos un nuevo nombre al archivo preexistente de configuración del directorio **/boot**, en nuestro entorno de la estructura del núcleo. Teclee

```
[root@serverA linux-2.6.12.1]# cp /boot/config-`uname -r` .config
```

En este caso, usamos **uname -r** para ayudarnos a obtener el archivo de configuración para el núcleo en ejecución. El comando **uname -r** imprime la emisión del núcleo en ejecución. Su uso aquí nos ayuda a garantizar que estamos obteniendo la versión exacta que queremos, sólo en caso que haya otras versiones presentes. En nuestro sistema Fedora de ejemplo, el comando es el equivalente de teclear en forma manual

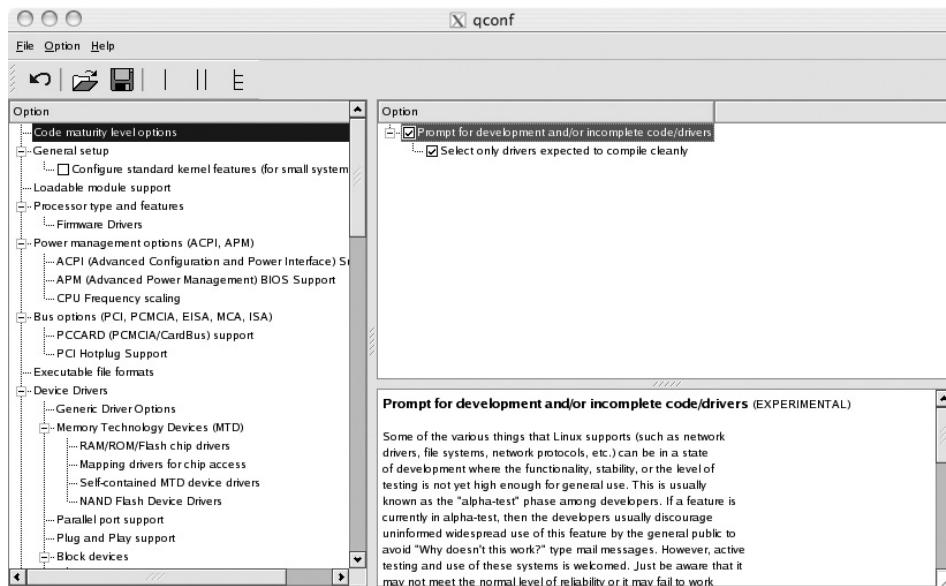
```
[root@serverA linux-2.6*]# cp /boot/config-2.6.11-1.1369_FC4 .config
```

NOTA El editor de configuración del núcleo de Linux busca de manera específica y genera un archivo nombrado **.config** en la raíz del árbol fuente del núcleo. Este archivo está escondido.

2. Lance la herramienta gráfica de configuración del núcleo. Teclee

```
[root@serverA linux-2.6.12.2]# make xconfig
```

Aparecerá una ventana semejante a ésta:



Si el comando antes dado se queja acerca de algunas dependencias faltantes, probablemente está diciendo que no cuenta con el entorno Qt de desarrollo apropiado. Suponiendo que está conectado a Internet, puede encargarse de su gimoteo usando Yum para instalar los paquetes apropiados sobre la Net, al teclear

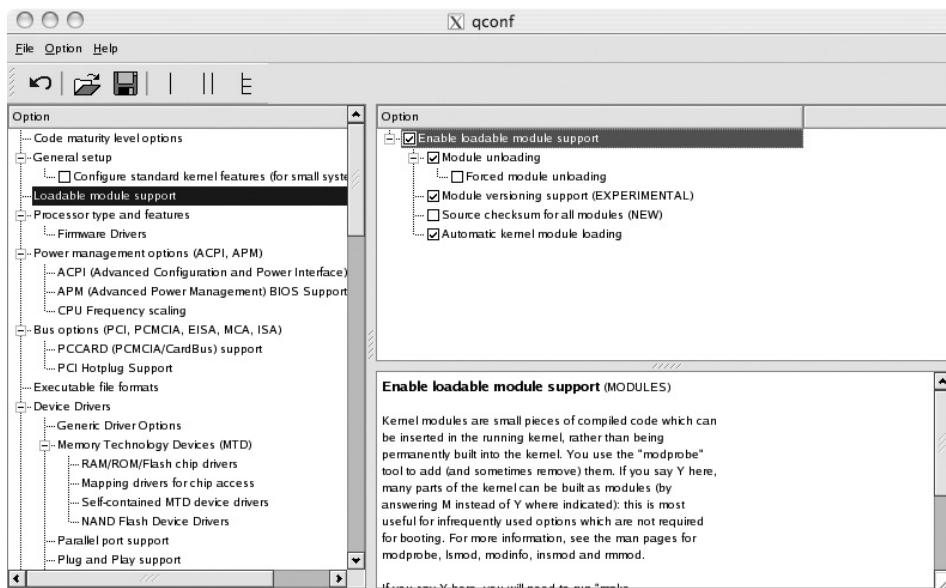
```
[root@serverA ~]# yum install qt-devel
```

O, en un sistema SuSE, use Yast para instalar las dependencias requeridas. Teclee

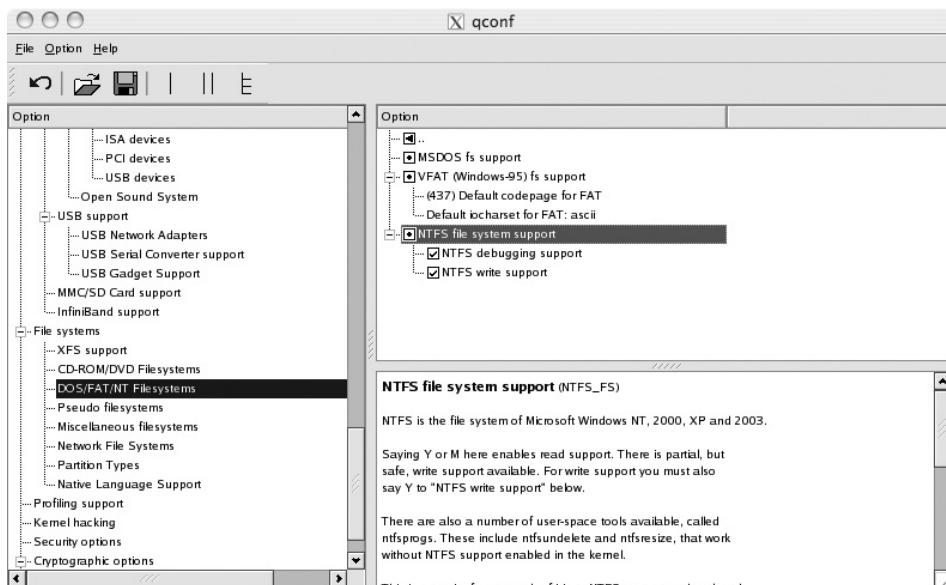
```
# yast -i qt3-devel
```

La ventana de configuración del núcleo (titulada qconf) que aparece está dividida en tres cuadros. En el de la izquierda se muestra una lista expansible con estructura de árbol de las opciones totales del núcleo que se pueden configurar. En el de arriba a la derecha, se presentan las opciones detalladas que se pueden configurar de la opción padre que en ese momento tiene el foco en el cuadro de la izquierda. Por último, en el de abajo a la derecha se presenta información de ayuda de la que se dispone para el ítem de configuración en ese momento seleccionado. En este cuadro se presenta información muy útil.

3. Examinemos ahora con un poco de más detalle una opción muy importante, seleccionándola en el cuadro de la izquierda. Use su ratón para hacer clic en el ítem Loadable Module Support (Soporte de módulos que se puede cargar). En casi todas las distribuciones de Linux verá que el soporte para esta característica está activado. En el cuadro de arriba a la derecha, seleccione la opción Enable Loadable Module Support y, a continuación, estudie la información de ayuda en línea que aparece en el cuadro de abajo a la derecha, como se muestra en la ilustración que sigue.



4. Enseguida, agregaremos soporte para el sistema de archivos NTFS en nuestro núcleo personal. En el cuadro de la izquierda, recorra la lista de secciones disponibles y, a continuación, seleccione la sección File Systems (Sistemas de archivos). Despues, en esa sección, seleccione DOS/FAT/NT Filesystems.
5. En el cuadro de arriba a la derecha, haga clic en el cuadro pequeño próximo a la opción NTFS File System Support, hasta que aparezca un pequeño punto en él. A continuación haga clic en los cuadros pequeños que están junto a las opciones NTFS Debugging Support (Soporte para corrección de errores en NTFS) y NTFS Write Support (Soporte de



Una nota rápida acerca de los módulos del núcleo

El soporte de módulos que se puede cargar es una característica del núcleo que permite la carga (o eliminación) dinámica de módulos del propio núcleo. Los módulos del núcleo son pequeñas piezas del código compilado que se pueden insertar dinámicamente en el núcleo en ejecución, en lugar de estar integrados de manera permanente en ese núcleo. De este modo, las características que no se usan con frecuencia se pueden desactivar pero no ocuparán espacio en la memoria cuando no se estén usando. Por fortuna, el núcleo puede determinar en forma automática qué cargar y cuándo cargarlo. Desde luego, no todas las características son elegibles para compilarse como módulo. El núcleo debe conocer unas cuantas cosas antes de que pueda cargar y descargar módulos, como la manera de tener acceso al disco duro y analizar el sistema de archivos en donde se almacenan los módulos susceptibles de ser cargados. También, es común referirse a algunos módulos del núcleo como controladores.

escritura en NTFS). Cuando haya terminado, en cada uno de ellos debe aparecer una marca de verificación, como las que se muestran enseguida:

NOTA Para cada opción, del cuadro de arriba a la derecha, un pequeño cuadro en blanco indica que la característica en cuestión está desactivada. Un pequeño cuadro con una marca de verificación indica que la característica está activada. Un pequeño cuadro con un punto indica que la característica se va a compilar como módulo. Haciendo clic en el cuadrito se recorrerá el ciclo por los tres estados.

6. Por último, guarde sus cambios en el archivo **.config** en la raíz de su árbol fuente del núcleo. Haga clic en File, en la barra de menús de la ventana qconf y seleccione la opción Save.

SUGERENCIA Para ver los resultados de los cambios que ha hecho usando la herramienta GUI de qconf, use la utilidad **grep** con el fin de ver el archivo **.config** que guardó. Teclee

```
[root@serverA linux-2.6.12.1]# grep NTFS .config
CONFIG_NTFS_FS=m
CONFIG_NTFS_DEBUG=y
CONFIG_NTFS_RW=y
```

7. Cuando haya terminado, cierre la ventana de qconf.

Compilación del núcleo

En la sección anterior, recorrimos paso a paso el proceso de creación de un archivo de configuración para el núcleo personal que queremos estructurar. En esta sección, ahora realizaremos la estructuración real del núcleo. Pero antes de hacer esto, agregaremos una sencilla personalización más al proceso completo.

La personalización final será agregar una pieza adicional de información usada en el nombre final de nuestro núcleo. Esto nos ayudará a poder diferenciar de manera absoluta este núcleo de cualquier otro con el mismo número de versión. Agregaremos la etiqueta “custom” a la información de la versión del núcleo. Esto se puede hacer editando el **Makefile** principal y anexando la etiqueta que queremos a la variable **EXTRAVERSION**.

Esta etapa de estructuración del núcleo es, con mucho, la más fácil, pero también lleva la mayor parte del tiempo. Todo lo que se necesita en este punto es ejecutar el comando **make**, el cual entonces generará en forma automática y se encargará de cualesquier aspectos de dependencias, compilará el propio núcleo y cualesquier características (o controladores) que se activarán como módulos que puedan cargarse.

Debido a la cantidad de código que es necesario compilar, esté listo para esperar al menos durante unos cuantos minutos; dependiendo del poder de procesamiento de su sistema.

Adentrémonos en los pasos específicos requeridos para compilar su nuevo núcleo:

1. En primer lugar, agregaremos una pieza adicional a la cadena de identificación para el núcleo que estamos a punto de estructurar. Mientras todavía se encuentra en la raíz del árbol fuente del núcleo, abra el **Makefile** para su edición con cualquier editor de textos. La variable que queremos cambiar se encuentra muy cerca de la parte superior del archivo. Cambie la línea del archivo que se mira como

```
EXTRAVERSION = .1
a
EXTRAVERSION = .1-custom
```

2. Guarde sus cambios en el archivo y salga del editor de textos.
3. El único comando que necesita aquí para compilar el núcleo es el **make**. Teclee

```
[root@serverA linux-2.6.12.1]# make
CHK      include/linux/version.h
SPLIT    include/linux/autoconf.h -> include/config/*
...<OUTPUT TRUNCATED>...
CC       sound/usb/usx2y/snd-usb-usx2y.mod.o
LD [M]   sound/usb/usx2y/snd-usb-usx2y.ko
```

4. El producto final de este comando (es decir, el núcleo) se está situando bonito y esperando en la trayectoria: <**kernel-source-tree**>/arch/i386/boot/bzImage.
5. Debido a que se compilaron porciones del núcleo como módulos (por ejemplo, el módulo NTFS), necesitamos instalar esos módulos. Teclee

```
[root@serverA linux-2.6.12.1]# make modules_install
```

En un sistema Fedora, con este comando se instalarán todos los módulos compilados del núcleo en el directorio **/lib/modules/<new_kernel-version>**. En este ejemplo, esta trayectoria se traducirá en el directorio **/lib/modules/2.6.12.1-custom/**. Ésta es la trayectoria desde la cual el núcleo cargará todos los módulos que puedan cargarse, según se necesiten.

Instalación del núcleo

De modo que, ahora, tiene un núcleo completamente compilado sólo esperando que se le instale. Es probable que el lector tenga un par de preguntas que hacer: precisamente *¿en dónde está el núcleo compilado?*, y *¿dónde diablos lo instalo?*

A la primera pregunta se le puede dar una respuesta fácil. Suponiendo que tiene una PC y está trabajando fuera del directorio `/usr/src/<kernel-source-tree>/`, el núcleo compilado que se creó en el ejercicio anterior se llamará `/usr/src/<kernel-source-tree>/arch/i386/boot/bzImage`, o bien, para ser precisos `/usr/src/linux-2.6.12.1/arch/i386/boot/bzImage`.

El archivo mapa correspondiente para esto estará ubicado en `/usr/src/<kernel-source-tree>/System.map`, o, como en nuestro ejemplo, `/usr/src/linux-2.6.12.1/System.map`. Necesitará los dos archivos para la fase de instalación.

El archivo **System.map** es útil cuando el núcleo se está comportando mal y generando mensajes de “Oops”. Un “Oops” se genera en algunos errores del núcleo. Se pueden deber a errores en el núcleo o a un hardware defectuoso. El error de “Oops” es semejante al de Blue Screen of Death (BSOD, Pantalla azul de muerte) en MS Windows. Estos mensajes contienen una gran cantidad de detalles acerca del estado en curso del sistema, incluyendo una gran cantidad de números hexadecimales. El **System.map** le da a Linux una posibilidad de convertir esos números en nombres legibles, haciendo más fácil la corrección de errores. Aunque esto es principalmente para el beneficio de los desarrolladores, puede ser práctico cuando el lector está informando de un problema.

Recorramos los pasos requeridos para instalar la nueva imagen del núcleo.

1. Mientras se encuentra en la raíz del directorio de estructura de su núcleo, copie el archivo **bzImage** en el directorio `/boot` y déle un nuevo nombre:

```
[root@serverA linux-2.6.12.1]# cp arch/i386/boot/bzImage \
 /boot/vmlinuz-<kernel-version>
```

en donde **kernel-version** es el número de versión del núcleo. Para el núcleo muestra que estamos usando en este ejercicio, el nombre de archivo sería **vmlinuz-2.6.12.1-custom**. De modo que el comando exacto para este ejemplo es

```
# cp arch/i386/boot/bzImage /boot/vmlinuz-2.6.12.1-custom
```

NOTA La decisión de dar el nombre de **vmlinuz-2.6.12.1-custom** a la imagen del núcleo es un tanto arbitraria. Es conveniente porque, por lo general, a las imágenes del núcleo se les menciona como **vmlinuz** y el sufijo del número de versión resulta útil cuando dispone de múltiples núcleos. Por supuesto, si quiere tener versiones múltiples del mismo núcleo (por ejemplo, uno con el soporte SCSI y el otro sin él), entonces necesitará diseñar un nombre más representativo. Por ejemplo, puede elegir un nombre como **vmlinuz-2.8.50-wireless**, para el núcleo de una laptop en la que se ejecuta Linux y que cuenta con capacidades inalámbricas.

2. Ahora que la imagen del núcleo está en su lugar, copie el archivo **System.map** correspondiente en el directorio `/boot` y déle otro nombre, aplicando la misma convención para asignar nombres. Teclee

```
[root@serverA linux-2.6.12.1]# cp System.map \
 /boot/System.map-2.6.12.1-custom
```

3. Con el núcleo en su lugar, el archivo **System.map** y los módulos en su lugar también, ahora estamos listos para dar el paso final. Teclee

```
[root@serverA linux-2.6.12.1]# new-kernel-pkg -v --mkinitrd --depmod \
 --install <kernel-version>
```

en donde ***kernel-version*** es el número de versión del núcleo. Para el núcleo muestra que estamos usando en este ejercicio, la versión del núcleo es 2.6.12.1-custom. De modo que el comando exacto para este ejemplo es

```
# new-kernel-pkg -v --mkinitrd --depmod --install 2.6.12.1-custom
```

El comando **new-kernel-pkg** usado aquí es un pequeño script shell muy elegante. Puede ser que no se disponga de él en todas las distribuciones de Linux, pero se encuentra en Fedora, RHEL y SuSE. Automatiza gran cantidad de las cosas finales que, de ordinario, hemos tenido que hacer en forma manual a fin de disponer el sistema para que inicialice el nuevo núcleo que acabamos de estructurar. En particular, hace lo siguiente:

- ▼ Crea la imagen apropiada de disco de la RAM inicial (la imagen initrd; es decir, el archivo **/boot/initrd-<kernel-version>.img**). El comando para hacer esto en forma manual en los sistemas en los que no se dispone de **new-kernel-pkg** es el **mkinitrd**.
- Ejecuta el comando **depmod** (con el cual se crea una lista de dependencias de módulos).
- ▲ Y, por último, actualiza la configuración del cargador de inicialización (en nuestro caso, actualiza el archivo **/boot/grub/grub.conf**).

La nueva entrada que se agregó en forma automática al archivo **grub.conf**, después de ejecutar el comando antes dado en nuestro sistema muestra, fue

```
title Fedora Core (2.6.12.1-custom)
root (hd0,0)
kernel /vmlinuz-2.6.12.1-custom ro root=/dev/VolGroup00/LogVol00 rhgb quiet
initrd /initrd-2.6.12.1-custom.img
```

NOTA Lo único que el comando **new-kernel-pkg** no hace es que en realidad no obliga a que el núcleo más reciente instalado inicialice en forma automática el núcleo predeterminado. Y, por consiguiente, puede ser que el lector tenga que seleccionar en forma manual, del menú del cargador de inicialización, el núcleo que quiere se inicialice, mientras el sistema está arrancando. Por supuesto, puede cambiar este comportamiento editando en forma manual el archivo **/boot/grub/grub.conf**, con el uso de cualquier editor de textos (vea el capítulo 6).

Inicialización del núcleo

La etapa que sigue es probar el nuevo núcleo con el fin de asegurarse que, en verdad, su sistema se puede inicializar con él.

1. Suponiendo que hizo todo de la manera exacta como lo prescribió el doctor y que todo funcionó de la manera exacta en que el mismo doctor dijo que lo haría, puede reiniciar con seguridad el sistema y seleccionar el nuevo núcleo en el menú del cargador de inicialización, en el curso del arranque. Teclee

```
[root@serverA ~]# reboot
```

- Después de que el sistema arranca, puede usar el comando **uname** para averiguar el nombre del núcleo actual. Teclee

```
[root@serverA ~]# uname -r  
2.6.12.1-custom
```

- El lector recordará que una de las características que agregamos a nuestro nuevo núcleo fue activar el soporte para el sistema de archivos NTFS. Asegúrese que el nuevo núcleo en efecto tiene soporte para el sistema de archivos NTFS, al presentar la información acerca del módulo NTFS. Teclee

```
[root@serverA ~]# modinfo ntfs  
filename:      /lib/modules/2.6.12.1-custom/kernel/fs/ntfs/ntfs.ko  
description:  NTFS 1.2/3.x driver - Copyright (c) 2001-2006 Anton  
Altaparmakov  
...<OUTPUT TRUNCATED>...  
vermagic:     2.6.12.1-custom 686 REGPARM gcc-4.0  
srcversion:   CFCABE9B21AEE7D9CA662A2
```

SUGERENCIA Suponiendo que en efecto tiene un sistema de archivos con formato NTFS, al que quiere tener acceso, puede cargar en forma manual el módulo NTFS al teclear

```
[root@serverA ~]# modprobe ntfs
```

El autor mintió: ¡No funcionó!

¿Dijo usted que el núcleo *no voló*? ¿Se congeló a la mitad de la inicialización? O bien, ¿se realizó la inicialización completa y, a continuación, nada funcionó correctamente? Antes que nada, *no entre en pánico*. Esta clase de problemas se le presentan a cualquiera, incluso a los profesionales. Después de todo, lo más probable es que se hayan presentado por intentar hacer algo con software no probado primero. De modo que no se preocupe, la situación es de lo más definitivamente reparable.

En primer lugar, advierta que se agregó una nueva entrada al archivo **/boot/grub/grub.conf** (o sea, al archivo **/boot/grub/menu.1st**) y *no se eliminó* la entrada anterior. De modo que puede usted caer con seguridad de regreso hacia el núcleo antiguo, que usted sabe que funciona, e inicializarse en él. Reinicie y, en el menú de GRUB, seleccione el nombre del núcleo anterior que se sabía que funcionaba. Esta acción debe llevarle de regreso a un estado conocido del sistema.

Ahora regrese a la configuración del núcleo y verifique que todas las opciones que seleccionó funcionarán para su sistema. Por ejemplo, ¿de manera accidental activó el soporte para el sistema de archivos Sun UFS, en lugar de para el sistema de archivos ext2 de Linux? ¿Fijó algunas opciones que dependían de que se fijaran otras opciones? Recuerde ver la pantalla informativa Help para cada opción del núcleo en la interfaz de configuración, asegurándose de que comprende qué hace cada opción y qué necesita usted hacerle para que funcione correctamente.

Cuando esté seguro que tiene correctos sus ajustes, recorra de nuevo los pasos del proceso de compilación y reinstale el núcleo. La creación de la imagen apropiada de disco de la RAM inicial (el archivo **initrd**) también es muy importante (vea la página man de **mkinitrd**). Si está ejecutando GRUB, sencillamente necesita editar el archivo **/boot/grub/menu.1st**, crear una entrada apropiada para su nuevo núcleo y, a continuación, reiniciar e intentar de nuevo.

No se preocupe, cada vez que compile un núcleo, lo hará mejor. Cuando cometa una equivocación, será más fácil regresar, hallarla y arreglarla.

PARCHADO DEL NÚCLEO

Como cualquier otro sistema operativo, Linux requiere mejoras periódicas para eliminar errores, mejorar el rendimiento y agregar nuevas características. Estas mejoras vienen en dos formas: en la forma de la publicación de un nuevo núcleo completo y en la de un parche. El núcleo nuevo completo funciona bien para las personas que no tienen por lo menos un núcleo completo ya descargado. Para los que tienen un núcleo completo ya descargado, los parches constituyen una mejor solución porque contienen sólo el código cambiado y, como tales, son mucho más rápidos de descargar.

Piense en un parche como comparable a un HotFix o paquete de servicios de Windows. Por sí mismo es inútil, pero cuando se agrega a una versión existente de Windows, obtendrá (al menos, es de esperar) un producto mejorado. La diferencia clave entre los HotFixes y los parches es que estos últimos contienen los cambios que es necesario hacer en el código fuente. Esto le permite a usted revisar los cambios en el código fuente, antes de aplicarlos. ¡Esto es mucho más bueno que adivinar si un arreglo descompondrá o no el sistema!

Puede averiguar acerca de los nuevos parches para el núcleo en muchos sitios de Internet. El sitio Web del vendedor de su distribución es un buen lugar para empezar; tendrá la lista no sólo de las actualizaciones sino también de los parches para otros paquetes. Una fuente importante es el Linux Kernel Archive oficial, en <http://www.kernel.org>. (Es decir, en donde obtuvimos el núcleo completo que usamos como ejemplo en la sección sobre instalación.)

En esta sección, demostraremos cómo aplicar un parche para actualizar una fuente del núcleo de Linux, versión 2.6.11, a la versión 2.6.12. El archivo exacto del parche que usaremos tiene el nombre de **patch-2.6.12.bz2**.

NOTA A veces podría usted ver los archivos de parches del núcleo con nombres como "patch-2.6.12-rc2.bz2", disponibles en el sitio Web kernel.org. En este ejemplo, el "rc2" que constituye parte del nombre y versión del parche (y, por consiguiente, de la versión final del núcleo), significa que el parche en cuestión es el parche "release candidate 2" (candidato de edición 2) que se puede usar para actualizar el árbol fuente apropiado del núcleo hacia el núcleo Linux versión 2.6.12-rc2. Lo mismo se cumple para un archivo de parche nombrado "patch-2.6-12-rc6.bz2", el cual será una "release candidate 6", y así sucesivamente.

Descarga y aplicación de parches

Los archivos de parches están ubicados en el mismo directorio del cual se descarga el núcleo. Esto se aplica a cada publicación principal de Linux; de modo que, por ejemplo, el parche para actualizar Linux versión 2.6.49 hacia Linux versión 2.6.50 puede estar ubicado en <http://www.kernel.org/pub/linux/kernel/v2.6/patch-2.6.50.bz2>. Los parches de prueba (o candidatos de edición puntual) están almacenados en el sitio Web kernel.org en el directorio **/pub/linux/kernel/v2.6/testing/**.

Cada nombre de archivo de parche tiene un prefijo con la cadena “patch” y un sufijo con el número de versión de Linux que se está instalando por medio del parche. Note que cada parche eleva a Linux hasta sólo en una versión; por tanto, el **patch-2.6.50** sólo se puede aplicar a linux-2-6-49. Por consiguiente, si tiene linux-2.6.48 y desea llevarlo hasta la versión 2.6.50, necesitará dos parches: **patch-2.6.49** y **patch-2.6.50**.

Los archivos de parches están almacenados en el servidor en un formato comprimido. En este ejemplo, estaremos usando el **patch-2.6.13.bz2** (de <http://www.kernel.org/pub/linux/kernel/v2.6/patch-2.6.13.bz2>). También necesitará la tarball actual de la fuente del núcleo que quiere actualizar. En este ejemplo, usaremos la fuente del núcleo que se descargó de <http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.12.tar.gz>.

Una vez que tenga los archivos del sitio kernel.org (o del espejo), muévalos hacia el directorio **/usr/src**. Supondremos que desempacó la fuente del núcleo que quiere actualizar, llevándola al directorio **/usr/src/linux-2.6.12**. A continuación, descomprimirá el parche usando la utilidad **bzip2** y, a continuación, mande por tubería la salida resultante hasta el programa patch, el cual entonces realizará el trabajo real de parchar/actualizar su núcleo.

1. Copie el archivo comprimido del parche que descargó, en un directorio en un nivel arriba de la raíz de su árbol fuente de su núcleo destino. Por ejemplo, suponiendo que el núcleo que quiere parchar se le ha deshecho la utilidad tar y enviado al directorio **/usr/src/linux-2.6.12/**, copiaría el archivo del parche en el directorio **/usr/src/**.
2. Cambie primero su directorio actual de trabajo hacia el nivel de arriba del árbol fuente del núcleo. En nuestro ejemplo, este directorio es el **/usr/src/linux-2.6.12/**. Teclee

```
[root@serverA ~]# cd /usr/src/linux-2.6.12/
```

3. Es una buena idea hacer una ejecución de prueba del proceso de parchado a fin de asegurarse que no hay errores y también para estar seguro que el nuevo parche en efecto se aplicará con limpieza. Teclee

```
[root@serverA linux-2.6.12]# bzip2 -dc ../patch-2.6.13.bz2 | patch -p1 --dry-run
```

4. Suponiendo que el comando que acaba de darse se ejecutó con éxito, sin errores, ahora está usted listo para aplicar el parche. Ejecute este comando para descomprimir el parche y aplicarlo a su núcleo:

```
[root@serverA linux-2.6.12]# bzip2 -dc ../patch-2.6.13.bz2 | patch -p1
```

en donde **../patch-2.6.13.bz2** es el nombre y la trayectoria hacia el archivo del parche. En su pantalla, se imprime una corriente de nombres de archivos. Cada uno de esos archivos ha sido actualizado por el archivo del parche. Si se tuvieron problemas con la actualización, se le informará aquí.

Si el parche funcionó...

Si el parche funcionó y usted no recibió errores, ¡casi ha terminado! Todo lo que necesita hacer es recompilar el núcleo. Sólo siga los pasos de la sección “Instalación del núcleo” que se encuentra antes en este capítulo.

Si el parche no funcionó...

Si tuvo errores en el curso del proceso de parchado del núcleo, no se desespere. Es probable que esto signifique una de dos cosas:

- ▼ No se puede aplicar el número de versión del parche al número de versión del núcleo (por ejemplo, trató de aplicar el **patch-2.6.50.bz2** al **Linux-2.6.60**).
- ▲ La propia fuente del núcleo ha cambiado. (¡Esto sucede a los desarrolladores que olvidan que hicieron cambios!)

La manera más fácil de componer cualquiera de estas dos situaciones es borrar el núcleo localizado en el directorio en donde lo desempacó y, a continuación, desempacar de nuevo el núcleo completo allí. Esto le garantizará tener un núcleo prístino. Después, aplique el parche. Es tedioso, pero si lo ha hecho una vez, es más fácil y rápida la segunda.

SUGERENCIA Por lo común, puede retirar (eliminar) cualquier parche que aplique, usando la opción **-R** con el comando **patch**. Por ejemplo, para retirar un parche versión 2.6.20 que se aplicó al núcleo Linux versión 2.6.19, mientras se encuentra en la raíz del árbol fuente del núcleo, teclearía

```
# bzip2 -dc ../patch-2.6.20.bz2 | patch -p1 -R
```

A veces, puede ser riesgoso retirar un parche y no siempre funciona; es decir, ¡el kilometraje de usted puede variar!

RESUMEN

En este capítulo, discutimos el proceso de configurar y compilar el núcleo de Linux. Éste no es exactamente un proceso trivial, pero realizarlo le da a usted el poder de tener un control de grano fino de su computadora que sencillamente no es posible con la mayor parte de los otros sistemas operativos. Compilar el núcleo es un proceso bastante directo. La comunidad de desarrollo de Linux ha suministrado herramientas excelentes que hacen el proceso tan indoloro como es posible.

Además de compilar los núcleos, recorrimos el proceso de actualizar los núcleos usando los parches de los que se dispone en el sitio Web Linux Kernel, <http://www.kernel.org>.

Cuando compile un núcleo por primera vez, hágalo con una máquina que no sea de producción, si es posible. Esto le da la oportunidad de tomar su tiempo y de enredarse con los muchos parámetros operacionales de los que se dispone. ¡Esto también significa que no irritará a sus usuarios si algo va mal!

Para los programadores curiosos de las entrañas del núcleo, existen muchas referencias en la forma de libros y sitios Web y, por supuesto, el propio código fuente es la última documentación.

CAPÍTULO 10



Perillas y carátulas:
el sistema
de archivos proc

La mayor parte de los sistemas operativos ofrece un mecanismo mediante el cual, cuando sea necesario, se puede sondear el interior del propio sistema así como fijar los parámetros operacionales. En Linux, este mecanismo lo suministra el sistema de archivos proc. Los sistemas operativos Windows de Microsoft permiten esto hasta cierto punto a través del Registry y Solaris lo hace a través de la herramienta **ndd** (Solaris también tiene un sistema de archivos proc). El directorio **/proc** es el punto de montaje para el sistema de archivos proc y, como consecuencia, a menudo se usan los dos términos de manera intercambiable. Con frecuencia, al sistema de archivos proc también se le menciona como el sistema de archivos virtuales.

En este capítulo, analizaremos el sistema de archivos proc y la manera en que funciona bajo Linux. Recorremos algunos panoramas generales y estudiaremos algunas entradas interesantes en **/proc** y, a continuación, demostraremos algunas tareas administrativas comunes con el uso de **/proc**. Finalizaremos con una breve mención del sistema de archivos del sistema (SysFS).

¿QUÉ SE ENCUENTRA DENTRO DEL DIRECTORIO /PROC?

Puesto que el núcleo de Linux es un componente tan clave en las operaciones del servidor, es importante que exista un método para intercambiar información con ese núcleo. De manera tradicional, esto se hace a través de las *llamadas del sistema*; funciones especiales escritas por los programadores para usarse al solicitar al núcleo que realice funciones en beneficio de ellos. Sin embargo, en el contexto de la administración de sistemas, las llamadas del sistema quieren decir que un programador necesita escribir una herramienta para que nosotros la usemos (a menos que, por supuesto, a usted le guste escribir sus propias herramientas). Cuando todo lo que usted necesita es hacer un simple ajuste pequeño o extraer alguna estadística del núcleo, tener que escribir una herramienta personal representa mucho más esfuerzo del que debe ser necesario.

Con el fin de mejorar la comunicación entre los usuarios y el núcleo, se creó el sistema de archivos proc. El sistema de archivos completo es en especial interesante porque en realidad no existe en alguna parte del disco; es sencillamente un resumen de la información del núcleo. Todos los archivos en el directorio corresponden a una función del núcleo o a un conjunto de variables que están en el núcleo.

NOTA Que proc sea un resumen no significa que no sea un sistema de archivos. Lo que en realidad significa es que tuvo que desarrollarse un sistema especial de archivos para tratar a proc de manera diferente que a los sistemas normales de archivos basados en el disco.

Por ejemplo, para ver un informe acerca del tipo de procesador que se encuentra en un sistema, podemos consultar uno de los archivos bajo el directorio **/proc**. El archivo en particular que contiene esta información es el **/proc/cpuinfo**. El archivo se puede ver con este comando:

```
[root@serverA ~]# cat /proc/cpuinfo
```

El núcleo creará en forma dinámica el informe, que muestre la información sobre el procesador y la manejará de regreso a **cat**, de modo que la podamos ver. Ésta es una manera sencilla y, sin embargo, poderosa con la que contamos para examinar el núcleo. El directorio **/proc** soporta una jerarquía de fácil lectura con el uso de subdirectorios y, como tal, es fácil hallar la información. Los directorios bajo **/proc** también están organizados de tal modo que los archivos que contienen información referente a temas semejantes están agrupados juntos. Por ejemplo, el directorio **/proc/scsi** ofrece informes en relación con el subsistema SCSI.

Incluso más que una ventaja es que el flujo de información toma los dos caminos: el núcleo puede generar informes para nosotros, y nosotros podemos pasar con facilidad información de regreso

al núcleo. Por ejemplo, la realización de un `ls -l` en el directorio `/proc/sys/net/ipv4` nos mostrará una gran cantidad de archivos que no son sólo de lectura, sino de lectura/escritura, lo cual quiere decir que algunos de los valores almacenados en esos archivos se pueden alterar al vuelo.

“¡Oiga! ¡La mayor parte de los archivos `/proc` tienen cero bytes y uno es ENORME! ¿Qué da?” No se preocupe si ha advertido todos esos archivos de cero bytes; la mayor parte de los archivos en el `/proc` son de cero bytes porque `/proc` en realidad no existe en el disco. Cuando el lector usa `cat` para leer un archivo `/proc`, el contenido de ese archivo se genera en forma dinámica por medio de un programa especial que se encuentra en el interior del núcleo. Como resultado, el informe nunca se guarda de regreso al disco y, por consiguiente, no ocupa espacio. Piense en ello a la misma luz de los scripts CGI para los sitios Web, en donde una página Web generada por un script CGI no se escribe de regreso al disco del servidor sino que se vuelve a generar cada vez que un usuario visita la página.

PRECAUCIÓN Ese único archivo enorme que ve en `/proc` es el `/proc/kcore`, el cual en realidad es un apuntador hacia el contenido de la RAM. Si usted tiene 128MB de RAM, el archivo `/proc/kcore` también tiene 128MB. Leer `/proc/kcore` es como leer el contenido en bruto de la memoria (y, por supuesto, no requiere permisos raíz).

Pequeños ajustes a los archivos dentro de `/proc`

Como se mencionó en la sección anterior, algunos de los archivos bajo el directorio (y subdirectorios) `/proc` tienen un modo de lectura-escritura. Examinemos uno de esos directorios con un poco de mayor profundidad. Los archivos que están en `/proc/sys/net/ipv4` representan parámetros en la pila TCP/IP que se pueden “afinar” en forma dinámica. Use el comando `cat` para mirar un archivo en particular y verá que la mayor parte de los archivos nada contienen más que un solo número. ¡Pero cambiando estos números, puede afectar el comportamiento de la pila TCP/IP de Linux!

Por ejemplo, el archivo `/proc/sys/net/ipv4/ip_forward` contiene un 0 (desactivado) de modo predeterminado. Esto le dice a Linux que no realice reenvíos IP cuando se tienen interfaces múltiples en la red. Pero si quiere estructurar algo como un encaminador Linux, necesita dejar que se realicen reenvíos. En esta situación, puede editar el archivo `/proc/sys/net/ipv4/ip_forward` y cambiar el número hacia 1 (activado).

Una manera rápida de hacer este cambio es mediante el uso del comando `echo` de este modo:

```
[root@serverA ~]# echo "1" > /proc/sys/net/ipv4/ip_forward
```

PRECAUCIÓN Tenga cuidado al hacer pequeños ajustes de los parámetros en el núcleo de Linux. No existe red de seguridad que lo salve de que realice ajustes erróneos para parámetros críticos, lo cual quiere decir que es del todo posible que pueda hacer que su sistema se caiga. Si no está seguro acerca de un ítem, es mejor que lo deje como está hasta que haya averiguado con toda certeza para qué es.

ALGUNAS ENTRADAS ÚTILES DE `/PROC`

En la tabla 10-1, se da una lista de las entradas de `/proc` que el lector puede encontrar útiles en la administración de su sistema Linux. Note que ésta es un grito lejano de lo que es una lista exhaustiva. Para obtener más detalles, lea con atención los directorios por sí mismo y vea qué encuentra. O también puede leer el archivo `proc.txt` en el directorio Documentation del código fuente del núcleo de Linux.

A menos que se diga otra cosa, el lector puede usar sencillamente el programa **cat** para ver el contenido de un archivo particular que esté en el directorio **/proc**.

Nombre de archivo	Contenido
/proc/cpuinfo	Información acerca de la CPU (o las CPU) en el sistema.
/proc/interrupts	Uso de IRQ en su sistema.
/proc/ioports	Presenta una lista de las regiones de puertos registrados que se usan para comunicación de entrada o salida (I/O) con los dispositivos.
/proc/iomem	Presenta el mapa actual de la memoria del sistema para cada dispositivo físico.
/proc/mdstat	Estado de la configuración RAID.
/proc/meminfo	Estado de uso de la memoria.
/proc/kcore	Este archivo representa la memoria física del sistema. A diferencia de los otros archivos bajo /proc , este archivo tiene un tamaño asociado con él. Su tamaño suele ser igual de la cantidad total de la RAM física disponible.
/proc/modules	La misma información producida como salida de lsmod .
/proc/pci	Informa de todos los dispositivos PCI conocidos en un sistema.
/proc/buddyinfo	La información almacenada en este archivo se puede usar para diagnosticar los aspectos de fragmentación de la memoria.
/proc/cmdline	Presenta los parámetros pasados al núcleo cuando éste arrancó (parámetros del momento de la inicialización).
/proc/swaps	Estado de las particiones de intercambio (swap), del volumen y/o de los archivos.
/proc/version	Número actual de la versión del núcleo, la máquina en la cual se compiló, y la fecha y la hora de la compilación.
/proc/ide/*	Información acerca de todos los dispositivos IDE.
/proc/scsi/*	Información acerca de todos los dispositivos SCSI.
/proc/net/arp	Tabla ARP (la misma salida que la de arp -a).
/proc/net/dev	Información acerca de cada uno de los dispositivos de la red (cuentas de paquetes, cuentas de error, etcétera).
/proc/net/snmp	Estadísticas SNMP acerca de cada protocolo.

Tabla 10-1. Entradas útiles bajo **/proc**

Nombre de archivo	Contenido
<code>/proc/net/sockstat</code>	Estadística sobre la utilización del enchufe de la red.
<code>/proc/sys/*</code>	Ajustes para la utilización del sistema de archivos por el núcleo. Muchos de éstos son valores que se pueden escribir; tenga cuidado acerca de cambiarlos, a menos que esté seguro de las repercusiones de hacerlo.
<code>/proc/sys/net/core/netdev_max_backlog</code>	Cuando el núcleo recibe paquetes de la red más rápido de como puede procesarlos, los coloca en una cola especial. De modo predeterminado, se permite un máximo de 300 paquetes en la cola. En circunstancias extraordinarias, puede ser que el lector necesite editar este archivo y cambiar el valor para el máximo permitido.
<code>/proc/sys/net/ipv4/icmp_echo_ignore_all</code>	Predeterminado = 0, lo que significa que el núcleo responderá a mensajes ICMP de respuesta de echo. La fijación de esto en 1 le dice al núcleo que cese de responder a esos mensajes.
<code>/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts</code>	Predeterminado = 0, lo que significa que el núcleo permitirá que se envíen respuestas ICMP a difusión o a direcciones de destinos múltiples.
<code>/proc/sys/net/ipv4/ip_forward</code>	Predeterminado = 0, lo que significa que el núcleo no reenviará entre las interfaces de la red. Para permitir el reenvío (por ejemplo, para encaminar), cambie esto a 1.
<code>/proc/sys/net/ipv4/ip_local_port_range</code>	Rango de los puertos que Linux usará cuando se origine una conexión. Predeterminado = 32768-61000.
<code>/proc/sys/net/ipv4/tcp_syn_cookies</code>	Predeterminado = 0 (desactivado). Cambie a 1 (activado) con el fin de activar la protección para el sistema contra ataques de inundación SYN.

Tabla 10-1. Entradas útiles bajo /proc (*cont.*)

Enumeración de entradas de /proc

Una lista del directorio de `/proc` revelará una gran cantidad de números de directorios cuyos nombres son sólo números. Estos números son las PID para cada proceso en ejecución en el sistema. Dentro de cada uno de los directorios de procesos se encuentran varios archivos que describen el estado del proceso y qué tanto de recursos está consumiendo éste (desde el punto de vista de un programador, los archivos de los procesos también son una manera fácil para que un programa obtenga información de sí mismo).

Por ejemplo, una lista larga de algunos de los archivos bajo **/proc** muestra

```
[root@serverA ~]# ls -l /proc
dr-xr-xr-x  3 root      root          0 Apr 10 04:12 1
dr-xr-xr-x  3 root      root          0 Apr 10 04:12 11068
dr-xr-xr-x  3 root      root          0 Apr 10 04:12 11117
...<OUTPUT TRUNCATED>...
```

Si mira de manera más detenida la carpeta con el nombre de “1” en la salida antes dada, advertirá que esta carpeta en particular representa la información acerca del proceso “init” (PID = 1). Una lista de los archivos bajo **/proc/1/** muestra

```
[root@serverA ~]# ls -l /proc/1/
dr-xr-xr-x  2 root      root          0 Apr 10 04:12 attr
-r-----  1 root      root          0 Apr 10 04:12 auxv
-r--r--r--  1 root      root          0 Apr 10 04:12 cmdline
...<OUTPUT TRUNCATED>...
lrwxrwxrwx  1 root      root          0 Apr 10 04:12 exe -> /sbin/init
```

Una vez más, como puede ver a partir de la salida, el archivo “/proc/1/exe” es un vínculo suave que apunta hacia el ejecutable actual para el programa **init** (**/sbin/init**).

La misma lógica se aplica a los otros directorios con nombres numéricos que están bajo **/proc**; es decir, representan procesos.

INFORMES Y AJUSTES COMUNES QUE SE HACEN CON PROC

Como ya se mencionó, el sistema de archivos proc es un sistema de archivos virtuales y, como resultado, los cambios a los ajustes predeterminados en **/proc** no sobreviven los reinicios. Si necesita que un cambio a un valor bajo **/proc** se fije en forma automática como / activado entre los reinicios del sistema, puede editar sus scripts de inicialización de modo que se haga el cambio en el momento del arranque, o bien, usar la herramienta **sysctl**. Por ejemplo, se puede usar el primer procedimiento para activar la funcionalidad de reenvío del paquete IP que está en el núcleo, cada vez que el sistema se inicialice. Para hacer esto, puede agregar la línea siguiente al final de su archivo **/etc/rc.d/rc.local**:

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

La mayor parte de las distribuciones de Linux ahora tienen una manera más graciosa de hacer persistente los cambios que se hagan al sistema de archivos proc.

En esta sección, veremos una herramienta que se puede usar para hacer cambios en forma interactiva en tiempo real a algunas de las variables almacenadas en el sistema de archivos proc.

Se usa la utilidad **sysctl** para presentar y modificar los parámetros del núcleo en tiempo real. Específicamente, se usa para afinar los parámetros que están almacenados bajo el directorio **/proc/sys** del sistema de archivos proc. Enseguida, se muestra un resumen de su uso y sus opciones:

```
sysctl [options] variable[=value]
```

en donde éstas son las opciones posibles:

Opciones	Explicación
variable [=value]	Usada para fijar o presentar el valor de una clave, en donde variable es la clave y value es el valor que se va a dar a esa clave. Por ejemplo, cierta clave se llama "kernel.hostname" y un valor posible para ella puede ser "serverA.example.com".
-n	Desactiva la impresión del nombre de la clave cuando se imprimen valores.
-e	Se usa esta opción para ignorar los errores acerca de claves desconocidas.
-w	Use esta opción cuando quiera cambiar un ajuste de sysctl .
-p <filename>	Carga los ajustes de sysctl provenientes del archivo especificado o de /etc/sysctl.conf , si no se da nombre de archivo.
-a	Presenta todos los valores de los que se dispone en ese momento.

Usaremos ejemplos reales para demostrar la manera de usar la herramienta **sysctl**. Los ejemplos demuestran unas cuantas de las muchas cosas que puede hacer con proc para complementar las tareas administrativas cotidianas. Los informes y las opciones susceptibles de cambiar, de las que se dispone a través de proc, son útiles en especial en las tareas relacionadas con la red. En los ejemplos también se proporciona alguna información básica relativa al ajuste de proc que queremos afinar.

Protección contra la inundación SYN

Cuando TCP inicia una conexión, lo primero que verdaderamente hace es enviar un paquete especial al destino, con la bandera fijada para indicar el inicio de una conexión. Esta bandera se conoce como la bandera SYN. El anfitrión destino responde con el envío de un paquete de reconocimiento de regreso a la fuente, llamado (con propiedad) SYNACK. Entonces el destino espera para que la fuente regrese un reconocimiento que muestre que ambos lados han quedado de acuerdo en los parámetros de su transacción. Una vez que se envían estos paquetes (a este proceso se le conoce como el "apretón de manos de tres vías"), los anfitriones fuente y destino pueden transmitir datos de uno a otro lado.

Debido a que es posible que múltiples anfitriones se pongan en contacto de manera simultánea con un solo anfitrión, es importante que el anfitrión destino se mantenga informado en todo momento de todos los paquetes SYN que le llegan. Las entradas SYN se almacenan en una tabla hasta que se completa el apretón de manos de tres vías. Una vez que se hace esto, la conexión sale de la tabla en la que se almacenó durante el apretón de manos SYN y se mueve hacia otra tabla que hace el seguimiento del rastro de las conexiones establecidas.

Se tiene una inundación SYN cuando un anfitrión fuente envía un gran número de paquetes SYN sin intención de responder al SYNACK. Esto conduce a un desborde de las tablas del anfitrión destino, haciendo de este modo que el sistema operativo se vuelva inestable. Es obvio que esto no es algo bueno.

Linux puede impedir las inundaciones SYN con el uso de una *syncookie*, un mecanismo especial que se encuentra en el núcleo y que se mantiene al tanto de la rapidez a la cual llegan los paquetes SYN. Si la syncookie detecta que la rapidez se está yendo por encima de cierto umbral,

empieza a desembarazarse agresivamente de las entradas que están en la tabla SYN y que no se mueven hacia el estado “establecido” dentro de un intervalo razonable. Una segunda capa de protección se encuentra en la propia tabla: si la tabla recibe una solicitud de SYN que haría que se desborde, esa solicitud se ignora. Esto significa que puede suceder que un cliente no pueda conectarse en forma temporal al servidor; pero también impide que se caiga por completo el servidor y saque a todos!

En primer lugar, use la herramienta **sysctl** a fin de presentar el valor actual para el ajuste **tcp_syncookie**. Teclee

```
[root@serverA ~]# sysctl net.ipv4.tcp_syncookies  
net.ipv4.tcp_syncookies = 0
```

La salida hace ver que, en este momento, este ajuste es desactivado (valor = 0). Para hacer que el soporte **tcp_syncookie** queda activado, introduzca este comando:

```
[root@serverA ~]# sysctl -w net.ipv4.tcp_syncookies=1  
net.ipv4.tcp_syncookies = 1
```

Debido a que las entradas de **/proc** no sobreviven a los reinicios del sistema, debe agregar la línea siguiente al final de su archivo de configuración **/etc/sysctl.conf**. Para llevar a cabo esto con el uso del comando **echo**, teclee

```
echo "net.ipv4.tcp_syncookies = 1" >> /etc/sysctl.conf
```

NOTA Por supuesto, en principio, debe asegurarse de que el archivo **/etc/sysctl.conf** no contiene ya una entrada para la clave que usted está tratando de afinar. Si es así, sencillamente puede editar a mano el archivo y cambiar el valor de la clave al valor nuevo.

Aspectos acerca de servidores de alto volumen

Como cualquier sistema operativo, Linux tiene recursos finitos. Si el sistema empieza a ejecutarse corto de recursos cuando está satisfaciendo solicitudes (como solicitudes de acceso a la Web), empezará a rechazar nuevas solicitudes de servicio.

La entrada **/proc/sys/fs/file-max** de **/proc** especifica el número máximo de archivos abiertos que Linux puede soportar en cualquier momento. El valor predeterminado en nuestro sistema Fedora era de 12094, pero éste se puede agotar con rapidez en un sistema muy ocupado, con una gran cantidad de conexiones de la red. Puede ser que se necesite elevarlo hasta un número más grande, como 20480. Usando una vez más el comando **sysctl**, teclee

```
[root@serverA ~]# sysctl -w fs.file-max=20480  
fs.file-max = 20480
```

No olvide anexar su cambio al archivo **/etc/sysctl.conf**, si desea que el cambio sea persistente.

Depuración de conflictos del hardware

Depurar los conflictos del hardware siempre es una tarea. Puede facilitar la carga mediante el uso de algunas de las entradas que se encuentran en **/proc**. Específicamente, las tres entradas siguientes para decirle lo que está pasando con su hardware:

- ▼ **/proc/pci** detalla todos los dispositivos PCI que estén en su sistema; es muy práctica cuando usted no conoce la fabricación y el modelo de un dispositivo, ¡y no quiere tener que abrir la máquina!
- **/proc/ioports** le dice las relaciones de los dispositivos con los puertos I/O (entrada/salida) y si no se tienen conflictos. Con el hecho de que los dispositivos PCI se están volviendo dominantes, esto no es un aspecto tan importante. Sin embargo, en tanto pueda comprar un nuevo tablero madre con ranuras ISA, siempre querrá contar con esta opción.
- ▲ **/proc/interrupts** le muestra la asociación de los números interrupt con los dispositivos de hardware. De nuevo, como con **/proc/ioports**, PCI está haciendo que esto sea menos de algo de qué preocuparse.

SysFS

SysFS (abreviatura de sistema de archivos del sistema) es semejante al sistema de archivos proc que se discutieron con anterioridad en este capítulo. Las semejanzas principales entre los dos son que los dos son sistemas de archivos virtuales y que ambos proporcionan un medio para que la información (en realidad, estructuras de datos) se exporten del interior del núcleo al espacio del usuario. SysFS suele montarse en el punto de montaje **/sys**.

Se puede usar el sistema de archivos SysFS para obtener información acerca de objetos del núcleo, como dispositivos, módulos, el bus del sistema, firmware, etcétera. Este sistema de archivos suministra una visión del árbol de dispositivos (entre otras cosas) según lo ve el núcleo. Esta vista presenta la mayor parte de los atributos conocidos de los dispositivos detectados, como el nombre del dispositivo, nombre del vendedor, clase PCI, recursos IRQ y DMA, y estado de la alimentación de la energía eléctrica. Algo de la información que usó al estar disponible (en las versiones del núcleo de Linux serie 2.4) bajo el sistema de archivos proc, ahora se puede hallar bajo SysFS.

Otra finalidad de SysFS es que proporciona una vista uniforme del espacio de dispositivos, lo que contrasta bruscamente con lo que en la actualidad se ve en el directorio **/dev**, el cual no cuenta con un esquema fijo de nombramiento, lo que permite a quien sea nombrar de cualquier manera a cualquier dispositivo. Los administradores familiarizados con Solaris se encontrarán en casa con las convenciones usadas de nombramiento. Sin embargo, la diferencia clave entre Solaris y Linux es que, bajo SysFS, las representaciones no proporcionan un medio para tener acceso al dispositivo a través del controlador del mismo. Para tener acceso basado en el controlador del dispositivo, los administradores necesitarán continuar usando la entrada **/dev** apropiada.

Una lista del nivel superior del directorio **sysfs** muestra estos directorios:

```
[root@serverA ~]# ls /sys/
block  bus  class  devices  firmware  kernel  module  power
```

Una mirada más profunda al directorio `/sys/devices` revela esta lista:

```
[root@serverA ~]# ls -l /sys/devices/
pci0000:00
platform
pnp0
pnp1
system
```

Si miramos una representación muestra de un dispositivo conectado al bus PCI de nuestro sistema, veremos estos elementos:

```
[root@serverA ~]# ls -l /sys/devices/pci0000:00/0000:00:00.0/
class
config
detach_state
device
driver
irq
local_cpus
resource
resource0
vendor
```

El elemento de más arriba bajo el directorio **devices** de la salida anterior describe el dominio PCI y el número del bus. En este caso, el bus particular del sistema es el bus PCI “`pci0000:00`”, en donde “`0000`” es el número del dominio y el número del bus es el “`00`”. Enseguida se da una lista de las funciones de los otros archivos:

Archivo	Función
class	Clase PCI
config	Espacio de configuración PCI
detach_state	Estado de conexión
device	Dispositivo PCI
irq	Número IRQ
local_cpus	Máscara de CPU cercana
resource	Direcciones de los anfitriones de recursos PCI
resource0 (resource0 . . . n)	Recurso PCI cero (o recurso PCI <i>n</i> , si está presente)
vendor	ID del vendedor del PCI (se puede hallar una lista de ID de los vendedores en el archivo <code>/usr/share/hwdata/pci.ids</code>)

RESUMEN

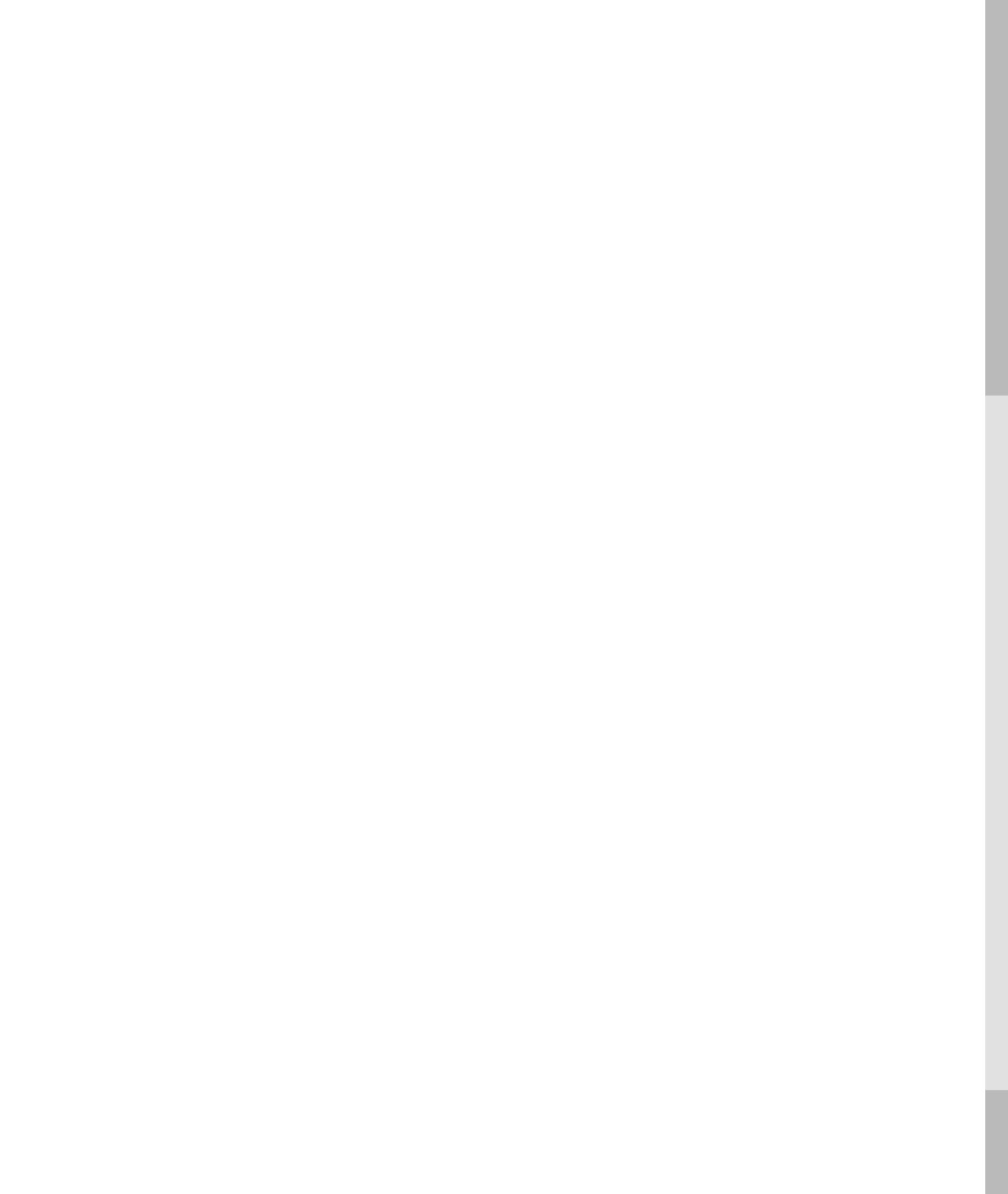
En este capítulo, aprendió acerca del sistema de archivos proc y la manera en que puede usarlo para echar un vistazo al interior del núcleo de Linux, así como para influir en la operación de ese núcleo. Las herramientas que se usan para llevar a cabo estas tareas son más o menos triviales (**echo** y **cat**), pero el concepto de un sistema de pseudoarchivos que no existen en el disco puede ser un poco difícil de captar.

Contemplando proc desde el punto de vista de un administrador de sistema, el lector aprendió a encontrar su camino en torno al sistema de archivos proc y la manera de obtener informes de diversos subsistemas (en particular el subsistema de operación en red). Aprendió cómo fijar los parámetros del núcleo para hacer frente a posibles mejoras futuras. Por último, también se hizo una breve mención del sistema de archivos virtuales SysFS.

PARTE III



Seguridad
y operación
en red



CAPÍTULO 11



TCP/IP para
administradores
de sistemas

Durante los últimos veinte años, una característica clave de UNIX ha sido la conciencia de red. Imaginar un sistema UNIX que no esté conectado a una red es como imaginar un carro deportivo sin una pista de carreras. Linux hereda ese legado y lo mantiene avanzando con toda su fuerza.

En la actualidad, ser un administrador de sistema es también tener una comprensión razonablemente fuerte de la red y de los protocolos usados para comunicarse sobre ella. Después de todo, si su servidor está recibiendo o enviando cualquier información, usted es el responsable de las acciones de ese servidor.

Este capítulo es una introducción a las entrañas del Transmission Control Protocol/Internet Protocol (Protocolo de control de la transmisión/protocolo de Internet), mejor conocido como TCP/IP. Abordaremos el contenido en dos partes: en primer lugar, recorreremos los detalles de los paquetes, Ethernet, TCP/IP y algunos detalles relacionados con el protocolo. Esta parte puede parecer un poco tediosa al principio, pero la perseverancia recibirá su recompensa en la segunda. En la segunda parte se presentan varios ejemplos de problemas comunes y cómo puede identificarlos con rapidez basándose en el conocimiento que acaba de adquirir del TCP/IP. A lo largo del camino, usaremos una herramienta maravillosa llamada **tcpdump**, herramienta que encontrará indispensable al final del capítulo.

Por favor, note que el intento de este capítulo no es ser un reemplazo completo de los muchos libros sobre TCP/IP, sino más bien una introducción desde el punto de vista de alguien que necesita preocuparse acerca de la administración de un sistema. Si desea una discusión más completa sobre TCP/IP, le recomendamos con vehemencia *TCP/IP Illustrated Vol. 1*, de Richard Stevens (Addison-Wesley, 1994).

LAS CAPAS

TCP/IP está estructurado en capas, de ahí las referencias a las *pilas* de TCP/IP. En esta sección echaremos una mirada a lo que son las capas de TCP/IP, su relación entre sí y, por último, por qué en realidad no se ajustan al modelo de siete capas Open Systems Interconnection (OSI, Interconexión de sistemas abiertos) de la International Organization for Standardization (ISO, Organización Internacional para la Estandarización; la sigla es de uso internacional porque también se le conoce como International Standards Organization). También traduciremos las capas OSI a significados que sean pertinentes para la red del lector.

Paquetes

En la parte más baja del sistema de capas se encuentra la unidad más pequeña de datos con la que a las redes les gusta trabajar: los *paquetes*. Éstos contienen tanto los datos que queremos transmitir entre nuestros sistemas como algo de información de control que ayuda al engranaje de las redes a determinar adónde debe ir el paquete.

NOTA Al discutir las redes, a menudo se intercambian los términos “paquete” y “marco”. En estas situaciones, las personas que se refieren a un marco a menudo quieren decir paquete. La diferencia es sutil. Un marco es el espacio en el que los paquetes van por una red. En el hardware, los marcos en una red están separados por preámbulos y postámbulos que le dicen al hardware en dónde empieza y en dónde termina un marco. Un paquete son los datos que están contenidos en el marco.

Un paquete TCP/IP típico que fluye en una red Ethernet se ve como se ilustra en la figura 11-1.

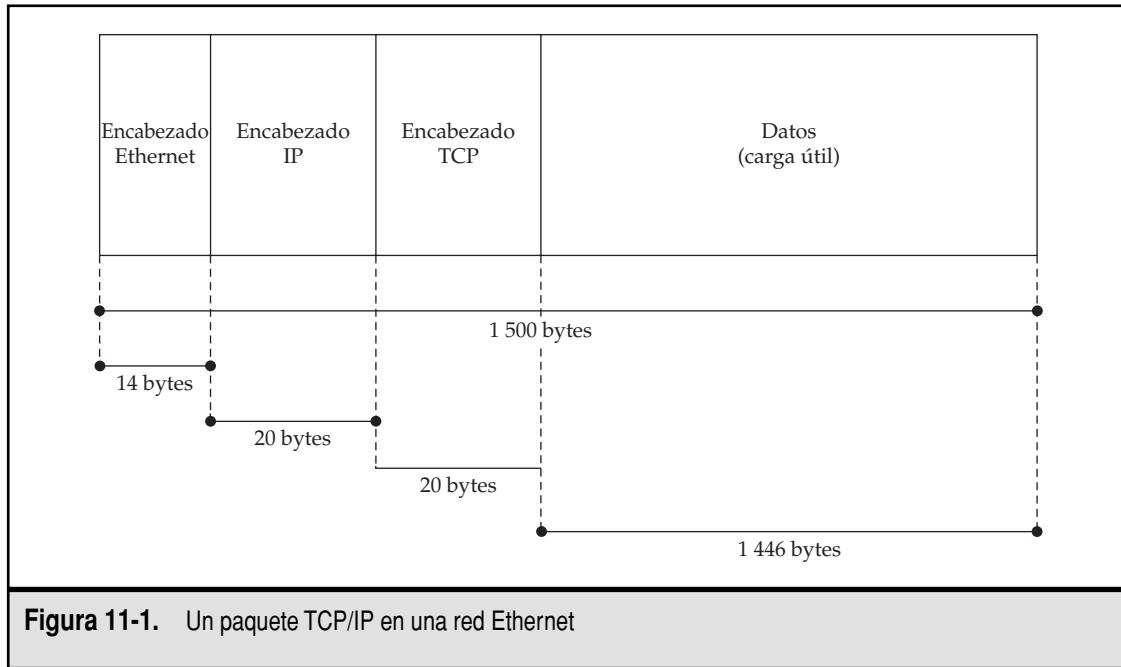


Figura 11-1. Un paquete TCP/IP en una red Ethernet

Marcos bajo Ethernet

Desde hace unos cuantos años, se ha actualizado la especificación de Ethernet para admitir marcos más grandes que 1 518 bytes. Estos marcos, llamados de manera apropiada marcos jumbo, pueden contener hasta 9 000 bytes. Esto, de modo conveniente, es espacio suficiente para un conjunto completo de encabezados TCP/IP, encabezados Ethernet, información NFS de control y una página de memoria (4K hasta 8K, dependiendo de la arquitectura de su sistema; Intel usa páginas de 4K). Debido a que los servidores ahora pueden empujar una página completa de memoria hacia fuera del sistema, sin tener que dividirla en paquetes diminutos, en algunas aplicaciones (como el servicio con disco remoto), ¡el rendimiento se puede ir hasta el techo!

El inconveniente de esto es que muy pocas personas usan marcos jumbo, de modo que el lector debe asegurarse de que las tarjetas de su red sean compatibles con sus commutadores, etcétera. Además, los marcos jumbo sólo se usan en los entornos Gigabit Ethernet. Linux soporta el uso de marcos jumbo con varios de los controladores Gigabit Ethernet, incluyendo acenic.o (conjunto de chips Tigon de Alteon, ahora parte de la familia Broadcom).

Un detalle importante: en las máquinas más antiguas, los marcos jumbo fueron necesarios para lograr una utilización de 100% de los enlaces de gigabits desde los servidores PC. Esto ya no es cierto. El rendimiento de las PC de la clase de servidor se ha elevado lo suficiente como para permitirles lograr rendimiento de gigabits en marcos de 1 518 bytes. Con más frecuencia que con ninguna, los cuellos de botella del rendimiento se originan en las limitaciones de las aplicaciones y no en limitaciones del núcleo o del hardware.

Como podemos ver en la figura 11-1, el protocolo divide en capas los paquetes; primero vienen las capas más bajas. En cada protocolo se usa un *encabezado* para describir la información necesaria para mover los datos de un anfitrión al siguiente. Los encabezados de paquetes tienden a ser pequeños; los encabezados para TCP, IP y Ethernet en su forma combinada más sencilla y más común sólo toman 54 bytes de espacio del paquete. Esto deja el resto de los 1 446 bytes del paquete para los datos.

En la figura 11-2 se ilustra cómo se pasa un paquete hacia arriba de una pila del protocolo. Veamos este proceso con un poco de más cuidado.

Cuando la tarjeta de redes del anfitrión recibe un paquete, en primer lugar hace una comprobación para ver si se supone que debe aceptar el paquete. Esto se hace al mirar las direcciones de destino ubicadas en los encabezados del paquete (más adelante, en este capítulo, hay más información acerca de los “encabezados”). Si la tarjeta de redes piensa que debe aceptar el paquete, conserva una copia de él en su propia memoria y genera una interrupción para el sistema operativo.

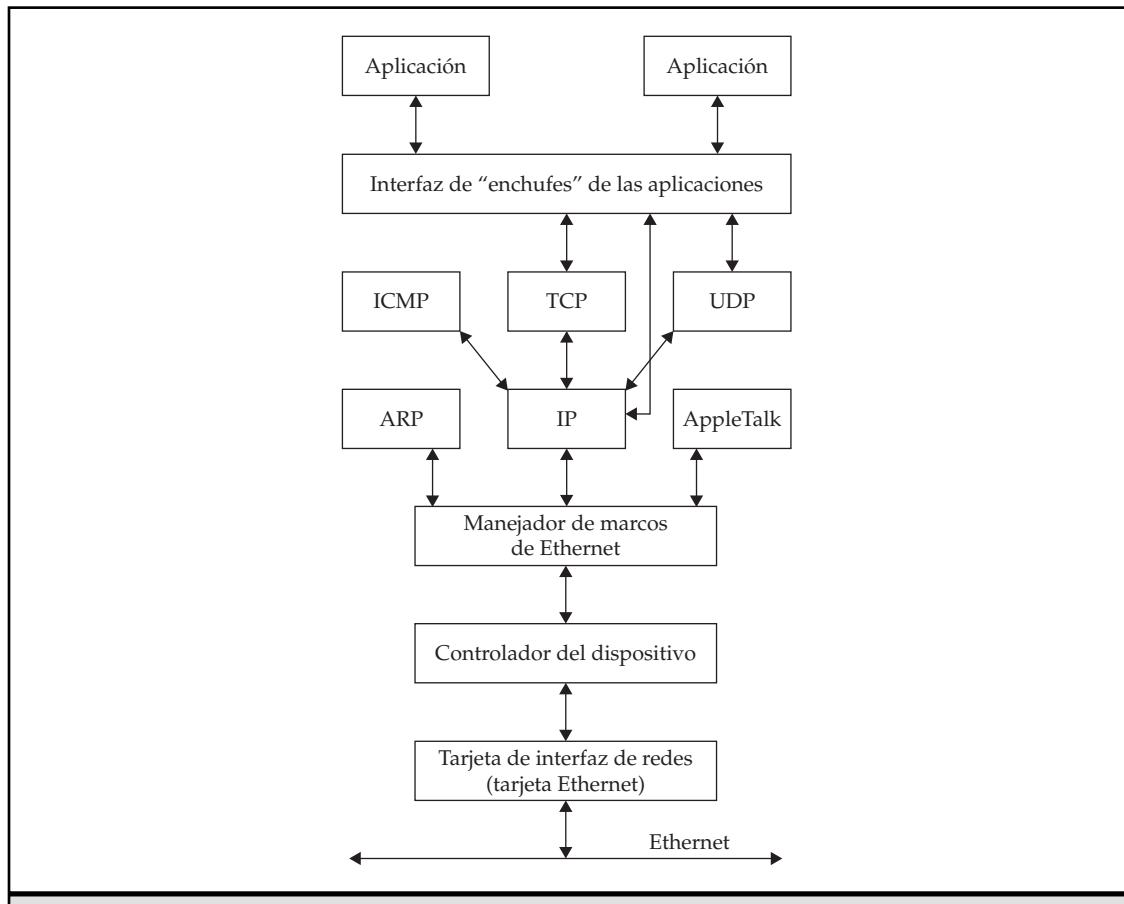


Figura 11-2. Trayectoria de un paquete a través de las redes Linux

Después de recibir esta interrupción, el sistema llama al controlador de dispositivos de la tarjeta de interfaz de redes (NIC) para procesar el nuevo paquete. El controlador de dispositivos copia el paquete de la memoria de la NIC en la memoria del sistema. Una vez que tiene una copia completa, puede examinar el paquete y determinar qué tipo de protocolo se está usando. Con base en el tipo de protocolo, el controlador manda una nota al manejador apropiado para ese protocolo, a fin de hacerle saber que tiene un paquete nuevo que procesar. El controlador de dispositivos entonces pone el paquete en un lugar en donde el software del protocolo (la “pila”) pueda hallarlo y regresa al proceso de interrupción.

Note que la pila no empieza a procesar el paquete de inmediato. Esto se debe a que el sistema operativo puede estar haciendo algo importante que necesita finalizar, antes de hacer que la pila procese el paquete. Ya que es posible que el controlador de dispositivos reciba muchos paquetes de la NIC con rapidez, existe una *cola* entre el controlador y el software de la pila. La cola sencillamente se mantiene al tanto del orden en el cual llegan los paquetes y las notas acerca de dónde están en la memoria. Cuando la pila está lista para procesar esos paquetes, los graba de la cola en el orden apropiado.

Conforme cada capa procesa el paquete, se eliminan los encabezados apropiados. En el caso de un paquete TCP/IP sobre Ethernet, el controlador desprenderá los encabezados de este último, IP arrancará el encabezado de él y TCP arrancará el que le corresponde. Con esto sólo quedarán los datos que necesitan ser entregados a la aplicación adecuada.

TCP/IP y el modelo OSI

El modelo de referencia OSI de ISO es uno bien conocido para describir las diversas capas de abstracción en las redes. El modelo OSI tiene siete capas (vea la figura 11-3) que, por desgracia, no se aplican muy bien al TCP/IP. Sin embargo, los vendedores de la red realizaron la administración necesaria para establecer una correspondencia y lograr una comprensión general de lo que representa cada capa de OSI en cada capa de TCP/IP que ha surgido.

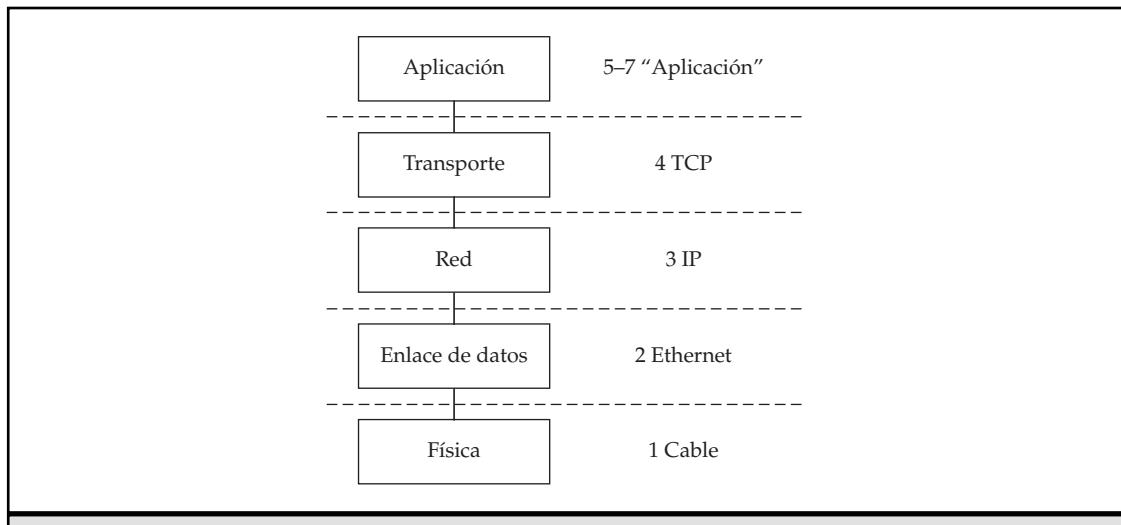


Figura 11-3. El modelo de referencia OSI

Capa 1

En la capa 1, la capa física, se describe el medio real en el cual fluyen los datos. En una infraestructura de redes, una pila de cable Cat 5 Ethernet y el protocolo de señalización se consideran la capa física.

Ethernet y la capa 2

La capa 2, la capa de enlace de datos, se usa para describir el protocolo de Ethernet. La diferencia entre la visión de OSI de la capa 2 y la Ethernet es que éste sólo se interesa en enviar los marcos y proporcionar una *suma de verificación* de ellos. La finalidad de la suma de verificación es permitir que el receptor valide si los datos llegaron como se enviaron. Esto se lleva a cabo al calcular la CRC del contenido del paquete y comparándola contra la suma de verificación que le proporcionó el remitente. Si al receptor le llega un marco corrupto (es decir, no coinciden las sumas de verificación) el paquete se deja caer aquí. Desde el punto de vista de Linux, no debe recibir un paquete que la tarjeta de interfaz de redes sepa que está corrupto.

Aunque en el modelo OSI se especifica de manera formal que la capa 2 debe manejar la retransmisión automática de un paquete corrupto, en Ethernet no se hace así. En lugar de ello, Ethernet se apoya en protocolos de nivel más elevado (en este caso, TCP) para manejar la retransmisión.

La responsabilidad primaria de Ethernet es simple: lleva el paquete de un anfitrión en una LAN (red de área local) hasta otro anfitrión en una LAN. Ethernet no tiene el concepto de una red global debido a las limitaciones en los tiempos de los paquetes, así como por el número de anfitriones que pueden existir en un solo segmento de la red. El lector se verá presionado para hallar más de 200, o algo así, de anfitriones en cualquier segmento dado, debido a los aspectos de ancho de banda y de simple administración. Es más fácil administrar grupos más pequeños de máquinas.

NOTA Ethernet se está usando cada vez más en redes de áreas metropolitanas (MAN, *metro area networks*) y en redes de área amplia (WAN, *wide area networks*) como un protocolo que sirve de marco para la conectividad. Aun cuando la distancia puede ser grande entre dos puntos extremos, estas redes no son de la Ethernet estándar del tipo de difusión que ve en un conmutador o un nodo central típicos. En lugar de ello, los vendedores de la red han optado por mantener la información sobre los marcos de la capa 2, como Ethernet, de modo que los routers no necesiten fragmentar los paquetes entre las redes. Desde el punto de vista de un administrador de sistema, no debe preocuparse si su proveedor de la red le dice que use Ethernet en su WAN/MAN; ¡ellos no han encadenado cientos de conmutadores para producir la distancia!

IP y la capa 3

El Protocolo de Internet (IP) es más sabio para el mundo en torno de él que Ethernet. IP entiende cómo comunicarse con los dos anfitriones dentro de la LAN inmediata, así como con anfitriones que están conectados con usted a través de routers (por ejemplo, anfitriones en otras subredes, la Internet, etcétera). Esto significa que un paquete IP puede recorrer su camino hacia cualquier otro anfitrión conectado a la misma red IP, en tanto exista una ruta hasta el anfitrión destino.

Como comparación, la capa 3 es la de la red. El IP no vive hasta la definición formal de la capa 3. La definición formal demanda control de las cuentas y de la congestión, ninguno de los cuales es capaz de realizar IP. Por supuesto, IP tratará de hacer llegar un paquete hasta su destino. Esto significa que los paquetes se pueden entregar fuera de orden, enviarse después de un largo

retraso o, incluso, dejarse caer a lo largo del camino. Además, IP sólo comprende la manera de hacer llegar un paquete hasta otro anfitrión. Una vez que un paquete llega al anfitrión, no existe información en el encabezado IP que le diga a cuál aplicación entregar los datos.

La razón por la que IP no proporciona más características que las de un simple protocolo de transporte es que se pretendió que fuera una base sobre la que se apoyaran otros protocolos. De los protocolos que usan IP, no todos necesitan conexiones confiables ni un orden garantizado de los paquetes. Por tanto, es responsabilidad de los protocolos de nivel superior proporcionar las características adicionales, si se necesitan.

TCP, UDP y la capa 4

TCP (Transmission Control Protocol, Protocolo de control de la transmisión) y UDP (User Datagram Protocol, Protocolo de datagramas de los usuarios) se aplican en la capa 4, la capa de transporte. TCP en realidad se aplica bastante bien a esta capa OSI, al proporcionar un transporte confiable para una *sesión*; es decir, una sola conexión del programa del cliente al programa del servidor. Por ejemplo, usando ssh para conectarse a un servidor se crea una sesión. El lector puede de tener múltiples ventanas ejecutando ssh desde el mismo cliente hacia el mismo servidor y, en cada caso, ssh tendrá su propia sesión.

Además de las sesiones, TCP también maneja el orden y la retransmisión de los paquetes. Si una serie de paquetes llega fuera de orden, la pila los pondrá de regreso en orden, antes de pasarlos hasta la aplicación. Si un paquete llega con cualquier clase de problema o va faltándole todo, TCP solicitará en forma automática al remitente que retransmita.

Por último, las conexiones TCP también son bidireccionales. Esto significa que tanto el cliente como el servidor pueden enviar y recibir datos en la misma conexión.

Como comparación, UDP no se aplica con tanta facilidad a OSI. Aun cuando UDP comprende el concepto de sesiones y es bidireccional, no suministra confiabilidad. En otras palabras, UDP no detectará la pérdida de paquetes o la duplicación de ellos, de la manera como lo hace TCP.

Después de todo, ¿por qué usar UDP?

Sin embargo, las limitaciones de UDP también constituyen su fuerza. UDP es una buena selección para dos tipos de tráfico: transacciones de solicitudes cortas/respuesta que se acomodan en un paquete (como DNS) y corrientes de datos que se ajustan mejor a brincar los datos perdidos y a seguir adelante (como corrientes de audio y video). En el primer caso, UDP es mejor porque una solicitud corta/respuesta por lo común no merece la carga general que TCP requiere para garantizar la confiabilidad. La aplicación suele ser más adecuada para agregar lógica adicional con el fin de retransmitir por su cuenta, en el caso de paquetes perdidos.

En el caso de datos en corriente, en realidad los desarrolladores no quieren la confiabilidad de TCP. Preferirían que simplemente se pasen por alto los paquetes perdidos, bajo la hipótesis (razonable) de que la mayor parte de los paquetes llegarán en el orden deseado. Esto se debe a que los escuchas/espectadores son mucho mejores en el manejo de interrupciones cortas en el audio (¡y se molestan menos!) que como son en los retrasos.

HTTP, SSL, XML y las capas 5 a 7

Técnicamente, las capas 5 a 7 de OSI tienen una finalidad específica, pero en la jerga de TCP/IP todas se amontonan en la capa de aplicación. Desde el punto de vista técnico, todas las aplicaciones en las que se usa TCP o UDP se asientan aquí; no obstante, en el mercado, a la capa 7 en general se le llama tráfico HTTP. Algunos vendedores se refieren a ella como la capa 5, pero desde nuestro punto de vista, las dos son lo mismo.

SSL es un tanto un pájaro raro, la cual, por lo común, no se asocia con alguna de las capas. Se asienta, con todo derecho, entre la capa 4 (TCP) y la 7 (la aplicación, por lo común HTTP) y se puede usar para codificar corrientes TCP arbitrarias. En general, SSL no se menciona como una capa. Sin embargo, el lector debe notar que SSL puede codificar conexiones TCP arbitrarias, no sólo HTTP. Muchos protocolos, como POP e IMAP, ofrecen SSL como una opción de codificación y el advenimiento de la tecnología SSL-VPN muestra cómo se puede usar SSL como un túnel arbitrario.

Los datos XML también pueden ser confusos. A la fecha, no existe protocolo para enmarcar para XML que se ejecute directamente arriba de TCP. En lugar de ello, en los datos XML se usan protocolos existentes, como HTTP, DIME y SMTP. (DIME se creó en forma específica para la transmisión de XML.) Para la mayor parte de las aplicaciones, en XML se usa HTTP, lo cual, desde un punto de vista de la división en capas, se mira como esto: Ethernet -> IP -> TCP -> HTTP -> XML. XML puede envolver otros documentos XML dentro de él mismo. Por ejemplo, SOAP puede envolver firmas digitales dentro de él. Para obtener información adicional sobre el propio XML, eche una mirada a <http://www.oasis.org> y a <http://www.w3c.org>.

ICMP

El Internet Control Message Protocol (ICMP, Protocolo de mensajes de control de Internet) se diseñó en especial para que un anfitrión se comunique con otro sobre el estado de la red. Ya que los datos sólo los usa el sistema operativo y no los usuarios, ICMP no soporta el concepto de números de puerto, entrega confiable ni orden garantizado de los paquetes.

Cada paquete ICMP contiene un *tipo* que le dice al destinatario cuál es la naturaleza del mensaje. El tipo más popular es "Echo-Request", el cual usa el infame programa ping. Cuando un anfitrión recibe el mensaje ICMP "Echo-Request", responde con un mensaje ICMP "Echo-Reply". Esto permite al remitente confirmar que el otro anfitrión se encuentra en actividad y, puesto que podemos ver cuánto tiempo tarda el mensaje en ser enviado y respondido, obtenemos una idea de la latencia de la red entre los dos anfitriones.

NOTA De vez en vez, el lector puede oír que se mencione la "capa 8". Por lo común, esto constituye una referencia al sarcasmo estadounidense. Lo normal es que se mencione como capa 8 a la capa "política" o "financiera", lo que significa que por encima de todas las redes se encuentran las personas. Y las personas, a diferencia de las redes, son no deterministas. Lo que puede tener sentido para la red, puede que no lo tenga políticamente y, de este modo, se construye una infraestructura aparentemente arbitraria. Un ejemplo sencillo: dos jefes de departamento dentro de la misma compañía rechazan reconocerse entre sí. Cuando averiguan que comparten la red, demandan tener su propia infraestructura (routers, comutadores, etc.) y estar colocados en redes diferentes; sin embargo, al mismo tiempo, son capaces de comunicarse entre sí: sólo a través de firewalls seguros. Lo que puede haber sido una red buena y simple (y funcional) ahora es mucho más compleja de lo que necesita ser, todo debido a la capa 8... (esto puede que suene como una caricatura de Dilbert, ¡pero recuerde que el creador de éste usa una gran cantidad de historias verdaderas de sus lectores!)

ENCABEZADOS

Al principio, en este capítulo, aprendimos que un paquete TCP/IP sobre Ethernet era una serie de *encabezados* para cada protocolo, seguida por los datos reales que se están enviando. Los “encabezados de paquetes”, como por lo general se les conoce, sencillamente son esas piezas de información que le dicen al protocolo cómo manejar el paquete.

En esta sección, echaremos una mirada a estos encabezados, usando la herramienta **tcpdump**. Todas las distribuciones comunes de Linux la tienen preinstalada; sin embargo, si quiere ver cuál es el estado más reciente de ella, puede visitar el sitio Web de **tcpdump** en <http://www.tcpdump.org>. Después de leer este capítulo, puede ser que el lector encuentre práctico leer la página completa de manual para **tcpdump**.

NOTA El lector debe ser raíz para ejecutar el comando **tcpdump**.

Ethernet

Ethernet tiene una historia muy interesante. Como resultado, existen dos tipos de encabezados Ethernet: 802.3 y Ethernet II. Por fortuna, aun cuando las dos se miran semejantes, existe una prueba sencilla para decirles que se separan. Empecemos por mirar el contenido del encabezado Ethernet (vea la figura 11-4).

El encabezado Ethernet contiene tres entradas: la dirección destino, la dirección fuente y el tipo de protocolo del paquete.

Las direcciones Ethernet, también conocidas como direcciones MAC (Media Access Control, Control de acceso a los medios); no existe relación con la Apple Macintosh, son números de 48 bits (seis bytes) que identifican de manera única cada tarjeta Ethernet en el mundo. Aun cuando es posible cambiar la dirección MAC de una interfaz, no se recomienda hacer esto, ya que la pre-determinada está garantizada que es única y todas las direcciones MAC en un segmento de una LAN deben ser únicas.

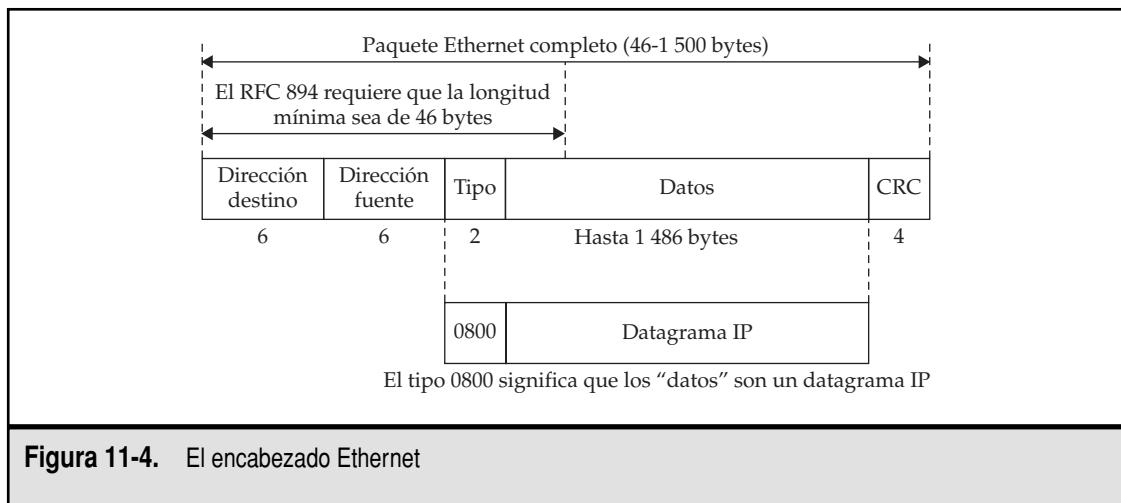


Figura 11-4. El encabezado Ethernet

NOTA Un paquete que se envía como una difusión (lo que significa que todas las tarjetas de la red deben aceptar este paquete) tiene la dirección destino fijada como ff : ff : ff : ff : ff : ff.

El tipo de protocolo del paquete es un valor de dos bytes que nos dice a cuál protocolo se le debe entregar este paquete en el lado del receptor. Para los paquetes IP, este valor es el hexadecimal (decimal 2048).

El paquete que se acaba de describir aquí es uno Ethernet II (lo normal es que sólo se le llame *Ethernet*). En los paquetes 802.3, las direcciones MAC del destino y la fuente permanecen en su lugar; sin embargo, los dos bytes que siguen representan la longitud del paquete. La manera en que usted puede decir la diferencia entre los dos tipos de Ethernet es que no hay tipo de protocolo con un valor de menos de 1 500. Por consiguiente, cualquier encabezado Ethernet en donde el tipo de protocolo es menor que 1 500 en realidad es un paquete 802.3. Considerándolo realista, es probable que el lector no vea más muchos paquetes 802.3 (si es que ve alguno).

Manera de ver los encabezados Ethernet

Para ver los encabezados Ethernet en su red, ejecute el comando siguiente:

```
[root@hostA ~]# tcpdump -e
```

Éste le dice a **tcpdump** que descargue los encabezados Ethernet junto con los encabezados TCP e IP.

Ahora genere algo de tráfico visitando un sitio Web, o bien, use ssh para comunicarse con otro anfitrión. Al hacerlo, generará una salida como ésta:

```
15:46:08.026966 0:d0:b7:6b:20:17 0:10:4b:cb:15:9f ip 191: hostA.ssh >
10.2.2.2.4769: P 5259:5396(137) ack 1 win 17520 (DF) [tos 0x10]

15:46:08.044151 0:10:4b:cb:15:9f 0:d0:b7:6b:20:17 ip 60: 10.2.2.2.4769 > hostA.
ssh: . ack 5396 win 32120 (DF)
```

El principio de cada línea es una impresión de la hora en la que se vio el paquete. Las dos entradas siguientes de la línea son las direcciones MAC de la fuente y del destino, respectivamente, para el paquete. En la primera línea, la dirección MAC de la fuente es 0:d0:b7:6b:20:17 y la del destino es 0:10:4b:cb:15:9f.

Después de la dirección MAC está el tipo de paquete. En este caso, **tcpdump** vio 0800, y en forma automática lo convirtió en **ip** para nosotros, de modo que fuera más fácil de leer. Si no quiere que **tcpdump** convierta los números en nombres para usted (lo que es práctico en especial cuando no está trabajando su resolución DNS), puede ejecutar

```
[root@hostA ~]# tcpdump -e -n
```

en donde la opción **-n** le dice a **tcpdump** que no realice la resolución del nombre. Las dos mismas líneas anteriores, sin resolución del nombre, se mirarían como esto:

```
15:46:08.026966 0:d0:b7:6b:20:17 0:10:4b:cb:15:9f 0800 191: 10.2.2.1.22 >
10.2.2.2.4769: P 5259:5396(137) ack 1 win 17520 (DF) [tos 0x10]

15:46:08.044151 0:10:4b:cb:15:9f 0:d0:b7:6b:20:17 0800 60: 10.2.2.2.4769 >
10.2.2.1.22: . ack 5396 win 32120 (DF)
```

Advierta que, en cada línea, el **ip** se convirtió en **0800**, el nombre del anfitrión **hostA** se convirtió en **10.2.2.1** y el número del puerto **ssh** se convirtió en **22**. Más adelante en este capítulo, en la sección “TCP”, discutiremos el significado del resto de las líneas.

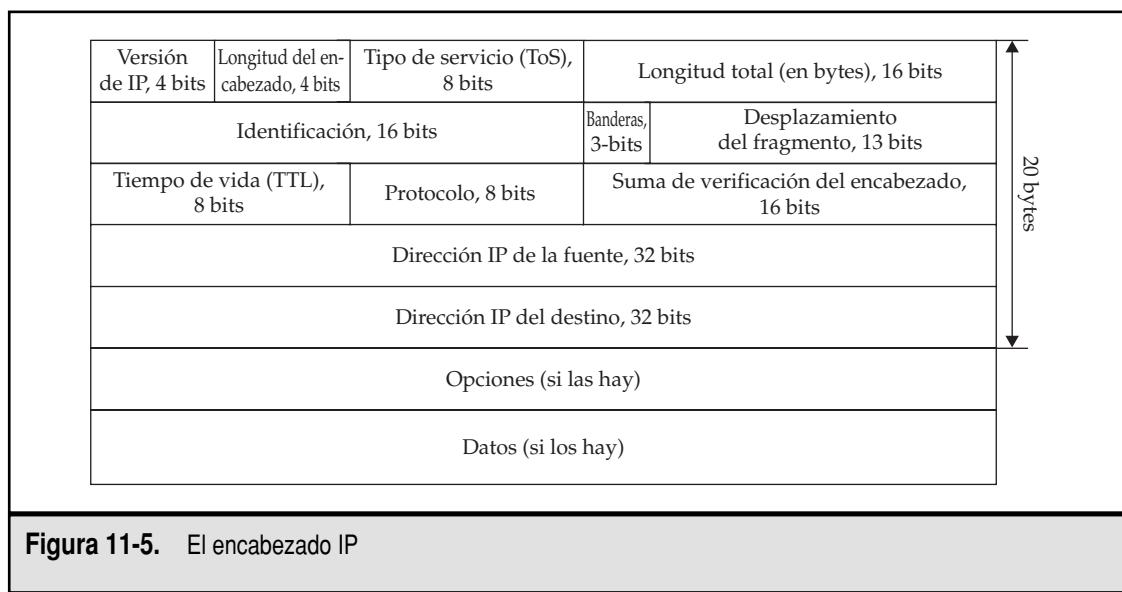
IP

Como se puede ver en la figura 11-5, el Internet Protocol tiene un encabezado ligeramente más complejo que Ethernet. Recorramos paso a paso lo que significa cada uno de los valores del encabezado.

El primer valor en el encabezado IP es el número de versión. La versión de IP que es de uso más común en la actualidad es la 4 (IPv4); sin embargo, en unos cuantos años más estará viendo más de la versión 6. Esta última versión ofrece muchas mejoras (y cambios) respecto a la 4, que lo mejor es dejar a los libros dedicados a la materia (de paso, la versión 5 fue una versión experimental diseñada para multimedia que nunca logró algo fuera del laboratorio).

El valor que sigue es la longitud del propio encabezado IP. Necesitamos saber cuán largo es el encabezado porque puede haber parámetros opcionales anexados al final del encabezado base. La longitud del encabezado nos dice cuántas opciones hay (si las hay). Para obtener el conteo de bytes de la longitud total del encabezado IP, multiplique este número por 4. Los encabezados IP típicos tendrán fijado el valor de la longitud del mismo en 5, lo que indica que hay 20 bytes en el encabezado completo.

El encabezado Type of Service (ToS) (tipo de servicio) le dice a las pilas IP qué clase de tratamiento se debe dar al paquete. En la fecha en que se está escribiendo esto, los únicos valores definidos son retraso minimizado, rendimiento maximizado, confiabilidad maximizada y costo minimizado. Para obtener más detalles, vea los RFC 1340 (<ftp://ftp.isi.edu/in-notes/rfc1340.txt>) y 1349 (<ftp://ftp.isi.edu/in-notes/rfc1349.txt>). Al uso de los bits del ToS a veces se le menciona como “coloración del paquete”; los usan los dispositivos para operación en red con el fin de conformación y determinación de prioridades de la velocidad.



El valor de la longitud total nos dice qué tan largo es el paquete completo, incluyendo los encabezados IP y TCP, pero sin incluir los encabezados Ethernet. Este valor se representa en bytes. Un paquete IP no puede ser más largo que 65 535 bytes.

Se supone que el número de identificación es un número único que usa un anfitrión con el fin de identificar un paquete en particular. Note que algunas herramientas de escaneo de seguridad identifican la “IP ID Randomization” como un aspecto posible de seguridad, pero esto no es del todo cierto. Si no se hace por completo aleatoria la IP ID, es posible usar eso como un medio para determinar el sistema operativo. Un atacante puede usar esa información para angostar la búsqueda para vectores posibles de ataque; sin embargo, la mayor parte de las pilas TCP/IP dan su identidad de varias maneras. No se enfoque con demasiada intensidad en este aspecto. Existen vectores de ataque más importantes de los que tiene usted que preocuparse.

Las banderas en el paquete IP nos dicen si el paquete está fragmentado o no. Se tiene fragmentación cuando un paquete IP es más grande que la *MTU* (unidad máxima de transmisión, *maximum transmission unit*) más pequeña entre dos anfitriones. La MTU define el paquete más grande que se puede enviar sobre una red en particular. Por ejemplo, la MTU de Ethernet es de 1 500 bytes. Por consiguiente, si se tiene un paquete IP de 4 000 bytes que se necesita enviar por Ethernet, dicho paquete se fragmentará en tres paquetes más pequeños. Los dos primeros serán de 1 500 bytes, y el último, de 1 000 bytes.

El valor de la desviación del fragmento nos dice cuál parte del paquete completo se está recibiendo. Continuando con el ejemplo del paquete IP de 4 000 bytes, la primera desviación del fragmento que recibiremos será de 0. La segunda será de 1 500 y la tercera de 3 000. La pila IP receptora tomará estos tres paquetes y los volverá a armar en un solo paquete grande, antes pasarlo hacia arriba de ella.

NOTA Ya no se tienen fragmentos IP con demasiada frecuencia sobre la Internet. Como consecuencia, muchos firewalls toman un enfoque verdaderamente paranoide acerca de tratar con los fragmentos IP, ya que son la fuente de ataques de negación del servicio (DoS).

La razón por la que los ataques DoS usan fragmentos IP es que muchas pilas IP no los manejan muy bien, lo cual provoca que se caiga el sistema. Linux maneja los fragmentos IP de modo correcto, lo cual lo convierte en un gran candidato para ser firewall. Si decide ir con un paquete firewall comercial, puede hallar como un buen ejercicio buscar una herramienta que puede enviar un gran número de fragmentos IP divididos y soltarlos sobre el firewall con el fin de verificar que manejará el ataque.

El campo de tiempo de vida (TTL, *time to live*) es un número entre 0 y 255 que significa cuánto tiempo se permite tener un paquete en la red antes de dejarlo caer. La idea que se encuentra detrás de esto es que, en el caso de un error de encaminamiento en donde el paquete está describiendo un círculo (también conocido como “bucle de encaminamiento”), el TTL haría que llegara el momento en que el paquete quedara fuera de tiempo y lo dejara caer, evitando de este modo que la red se llegue a congestionar por completo con paquetes viajando en círculo. A medida que cada router procesa el paquete, se hace disminuir el valor del TTL en uno. Cuando el TTL llega a cero, el router en el cual sucede esto envía un mensaje a través del protocolo ICMP (vea acerca del ICMP antes en el capítulo), informando al remitente de esto.

NOTA La capa 2 conmuta, no hace decrecer el TTL, sólo lo hacen los routers. La detección de bucles de los commutadores de la capa 2 no se apoya en el etiquetado de los paquetes sino que, por el contrario, utiliza el protocolo propio de los commutadores para comunicarse con otros commutadores de capa 2 con el fin de formar un “árbol de intervalos”. En esencia, un commutador de capa 2 establece una correspondencia con todos los commutadores adyacentes y envía paquetes de prueba (unidades puente de datos del protocolo, BPDU) y busca los paquetes de prueba generados por sí mismo. Cuando un commutador ve que uno de los paquetes le regresa, se encuentra un bucle y el puerto ofensor se cierra automáticamente al tráfico normal. Las pruebas se ejecutan en forma constante, de modo que si la topología cambia o falla la trayectoria primaria para un paquete, los puertos que se cerraron al tráfico normal se pueden volver a abrir.

El campo de protocolo en el encabezado IP nos dice a cuál protocolo de nivel superior se debe entregar este paquete. Por lo común, éste tiene un valor para TCP, UDP o ICMP. En la salida de **tcpdump** hemos visto que es este valor el que determina si la salida se lee como **udp** o **tcp**, después de presentar la combinación fuente y destino IP / puerto.

El último valor pequeño en este encabezado IP es la suma de verificación. Este campo contiene la suma de todos los bytes en el encabezado IP, incluyendo cualesquiera opciones. Cuando un anfitrión estructura un paquete IP para enviar, calcula la suma de verificación IP y la coloca en este campo. Entonces el receptor puede realizar el mismo cálculo y comparar los valores. Si los valores no coinciden, el receptor sabe que el paquete se corrompió en el curso de la transmisión (por ejemplo, la caída de un rayo que crea una perturbación eléctrica podría hacer que el paquete se corrompa).

Por último, los números que importan más en un encabezado IP: las direcciones IP de la fuente y el destino. Estos valores se almacenan como enteros de 32 bits, en lugar de la notación con puntos decimales que es más legible para los humanos. Por ejemplo, en lugar de 192.168.1.1, el valor sería el hexadecimal c0a80101 o el decimal 3232235777.

tcpdump e IP

De modo predeterminado, **tcpdump** no descarga todos los detalles del encabezado IP. Para ver todo, necesita especificar la opción **-v**. El programa **tcpdump** continuará presentando todos los paquetes que correspondan hasta que oprima **CTRL-C** para detener la salida. Puede pedir a **tcpdump** que se detenga automáticamente, después de un número fijo de paquetes, con el parámetro **-c** seguido por el número de paquetes por buscar. Por último, en beneficio de la brevedad, puede eliminar la impresión de la hora usando el parámetro **-t**. Suponiendo que queremos ver los dos paquetes IP siguientes, sin decodificación DNS, usaríamos los parámetros siguientes:

```
[root@hostA:~]# tcpdump -v -t -c 2 ip
68.121.105.169 > 68.121.105.170: icmp: echo request (ttl 127, id 21899, len 60)
68.121.105.170 > 68.121.105.169: icmp: echo reply (ttl 64, id 35004, len 60)
```

En la salida, vemos un paquete ping enviado y regresado. El formato de esta salida es

```
src > dest: [deeper protocols] (ttl, id, length)
```

en donde *src* y *dest* se refieren a la fuente y el destino del paquete, respectivamente. Para los paquetes TCP y UDP, la fuente y el destino incluirán el número del puerto después de la dirección IP. El extremo de cola de la línea muestra el TTL, la IP ID y la longitud, respectivamente. Sin la opción **-v**, el TTL sólo se muestra cuando es igual a 1.

TCP

El encabezado TCP es semejante a la IP en el sentido de que empaca bastante poca información en un espacio muy pequeño. Empecemos por mirar la figura 11-6.

Las dos primeras piezas de información en un encabezado TCP son los números de los puertos fuente y destino. Debido a que éstos sólo son valores de 16 bits, su rango es de 0 hasta 65535. Por lo común, el puerto fuente es un valor mayor que 1024, ya que los puertos 1 a 1023 se reservan para uso del sistema en la mayor parte de los sistemas operativos (incluyendo Linux, Solaris y las muchas variantes de MS Windows). Por otra parte, lo normal es que el puerto destino sea bajo; la mayor parte de los servicios populares residen allí, aunque éste no es un requisito.

En la salida de **tcpdump** vemos los números de los puertos inmediatamente después de la dirección IP. Por ejemplo, en la salida de **tcpdump -n -t**

```
192.168.1.1.2046 > 192.168.1.12.79: . 1:1(0) ack 1 win 32120 (DF)
```

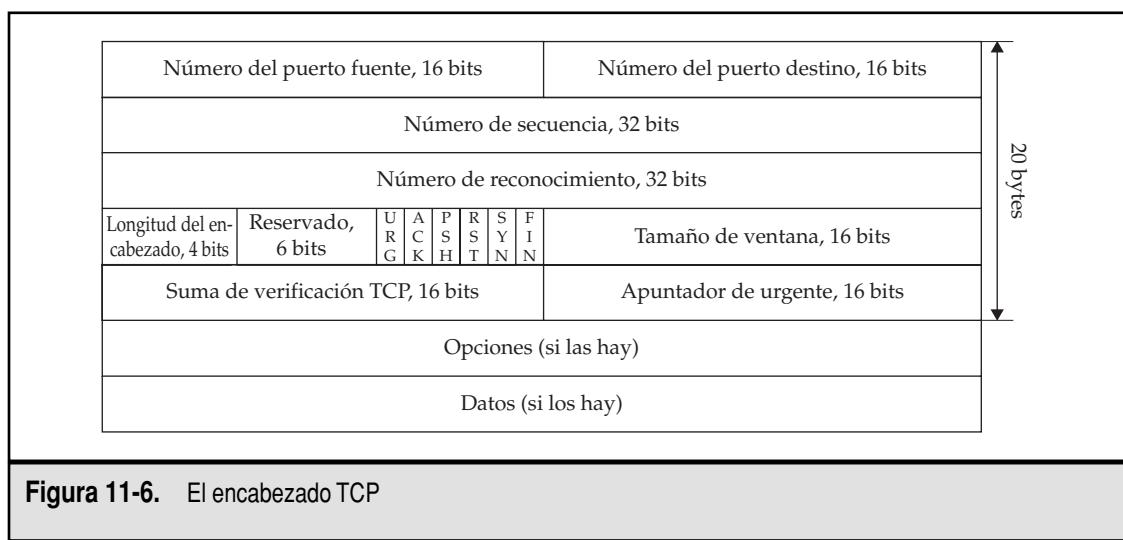
el número del puerto de origen es 2046, y el de destino es 79.

Los dos números siguientes en el encabezado TCP son los números de secuencia y reconocimiento. Estos valores los usa TCP para garantizar que el orden de los paquetes es el correcto y hacerle saber al remitente cuáles paquetes se han recibido de manera apropiada. En las tareas cotidianas de administración, el lector no debe tratar con ellos.

En la salida de **tcpdump** vemos números de secuencia en los paquetes que contienen datos. El formato es *número de inicio:número de finalización*. Miremos la siguiente salida **tcpdump** de **tcpdump -n -t**:

```
192.168.1.1.2046 > 192.168.1.12.79: P 1:6(5) ack 1 win 32120 (DF)
```

Vemos que los números de secuencia son 1:6, lo que significa que los datos se iniciaron en el número de secuencia 1 y finalizaron en el 6. En el número entre paréntesis que está inmediatamente después de los números de secuencia, podemos ver la longitud de los estados que se están enviando (cinco bytes, en este ejemplo).



En esta salida muestra también vemos el número de reconocimiento. Siempre que el paquete tenga fija la bandera de reconocimiento, puede ser usada por el receptor para confirmar cuántos datos han sido recibidos del remitente (vea la discusión de la bandera ACK más adelante en esta sección). **tcpdump** imprime **ack** seguido por el número de reconocimiento cuando ve un paquete con el bit de reconocimiento fijo. En este caso, el número de reconocimiento es 1, lo que significa que 192.168.1.1 está reconociendo el primer byte enviado a él por 192.168.1.12 en la conexión en curso.

NOTA Con el fin de hacer que la salida sea más legible, **tcpdump** usa valores relativos. Por tanto, un número de secuencia de 1 en realidad significa que los datos contenidos en el paquete es el primer byte que se envía. Si quiere ver el número real de secuencia, use la opción **-s**.

Semejante a la longitud del encabezado de IP, la longitud del encabezado de TCP nos dice lo largo que es ese encabezado, incluyendo cualesquiera opciones TCP. Cualquiera que sea el valor en el campo de longitud del encabezado, se multiplica por 4 para obtener el valor en bytes.

La parte siguiente es un poco difícil. En TCP se usa una serie de banderas para indicar si supone que el paquete inicia una conexión, contiene datos o termina una conexión. Las banderas (en el orden en que aparecen) son Urgent (URG, urgente), Acknowledge (ACK, reconocimiento), Push (PSH, presionar), Reset (RST, reinicializar), Synchronize (SYN, sincronizar) y Finish (FIN, terminar). Sus significados son los siguientes:

Bandera	Significado
URG	Mensaje de que en el paquete se encuentran datos urgentes que deben recibir procesamiento con prioridad. No existe una razón válida para que en los sistemas operativos modernos se use este bit.
ACK	Reconocimiento de que los datos se recibieron con éxito.
PSH	Solicitud de procesar de inmediato cualesquiera datos recibidos.
RST	Termina en forma inmediata la conexión.
SYN	Solicitud de iniciar una nueva conexión.
FIN	Solicitud para finalizar una conexión.

Por lo general, estas banderas se usan en combinación con otra. Por ejemplo, es común ver PSH y ACK juntas. Al usar esta combinación, en esencia el remitente le dice al receptor dos afirmaciones:

- ▼ Hay datos en este paquete que es necesario procesar.
- ▲ Estoy reconociendo que he recibido con éxito datos provenientes de usted.

El lector puede ver de inmediato cuáles banderas están en un paquete en la salida de **tcpdump**, después de la dirección IP de destino y del número de puerto. Por ejemplo,

```
192.168.1.1.2046 > 192.168.1.12.79: P 1:6(5) ack 1 win 32120 (DF)
```

En la línea anterior vemos que la bandera es P, por PSH. En **tcpdump** se usa el primer carácter del nombre de la bandera para indicar la presencia de ella (como S para SYN y F para FIN). La única excepción es ACK, la cual, en realidad, se pone como **ack**, más adelante en la línea (si el paquete sólo tiene fijado el bit de ACK, se usa un punto como sustituto en donde suelen imprimirse las banderas). ACK es una excepción porque hace que sea más fácil hallar cuál es el número de reconocimiento para ese paquete (vea el análisis sobre números de reconocimiento con anterioridad en esta sección; discutiremos las banderas con mayor detalle cuando veamos el establecimiento y la ruptura de una conexión).

La entrada que sigue en el encabezado es el tamaño de ventana. En TCP se usa una técnica llamada *ventana deslizante*, la cual permite a cada lado de una conexión decirle al otro cuánto espacio de memoria intermedia tiene disponible para tratar con las conexiones. Cuando llega un paquete nuevo en una conexión, el tamaño disponible de ventana disminuye en el tamaño de ese paquete, hasta que el sistema operativo tiene posibilidad de mover los datos de la memoria intermedia de entrada de TCP hacia el espacio de memoria intermedia de la aplicación receptora. El tamaño de ventana se calcula en términos de conexión por conexión. Veamos como ejemplo alguna salida de **tcpdump -n -t**:

```
192.168.1.1.2046 > 192.168.1.12.79: . 6:8(2) ack 1 win 32120 (DF)
192.168.1.12.79 > 192.168.1.1.2046: . 1:494(493) ack 8 win 17520 (DF)
192.168.1.1.2046 > 192.168.1.12.79: . 8:8(0) ack 495 win 31626 (DF)
192.168.1.1.2046 > 192.168.1.12.79: . 8:8(0) ack 495 win 32120 (DF)
```

En la primera línea podemos ver que 192.168.1.1 le está diciendo a 192.168.1.12 que, en ese momento, tiene 32 120 bytes disponibles en su memoria intermedia para esta conexión en particular. En el segundo paquete, 192.168.1.12 envía 493 bytes a 192.168.1.1 (al mismo tiempo, 192.168.1.12 le dice a 192.168.1.1 que su ventana disponible es de 17 520 bytes). 192.168.1.1 responde a 192.168.1.12 con un reconocimiento, diciéndole que ha aceptado en forma apropiada todo hasta el byte 495 de la corriente, lo cual en este caso incluye todos los datos que han sido enviados por 192.168.1.12. También está reconociendo que su ventana disponible ahora es de 31 626, lo cual es exactamente el tamaño original de ventana (32 120) menos la cantidad de datos que han sido recibidos (493 bytes). Unos cuantos instantes más tarde, en la cuarta línea, 192.168.1.1 le envía una nota a 192.168.1.12 expresando que ha transferido con éxito los datos a la memoria intermedia de la aplicación y que su ventana regresa a 32 120.

¿Un poco confuso? No se preocupe demasiado acerca de ello. Como administrador de sistema, no debe tratar con este nivel de detalle, pero resulta útil saber qué significan los números.

NOTA El lector puede haber advertido un error de uno en mis matemáticas en este caso: 32 120 – 493 es 31 627, no 31 626. Esto tiene que ver con los matices de los números de secuencia, los cálculos del espacio disponible, etcétera. Para tener conocimiento de toda la fealdad de la manera como funcionan las matemáticas, lea RFC 793 (<ftp://ftp.isi.edu/in-notes/rfc793.txt>).

El elemento siguiente en el encabezado TCP es la suma de verificación. Ésta es semejante a la suma de verificación IP en el sentido de que su finalidad es proporcionar al receptor una manera de verificar que los datos recibidos no están corruptos. A diferencia de la suma de verificación IP, la TCP en realidad toma en cuenta tanto el propio encabezado como los datos que se están enviando (técnicamente, también incluye el pseudoencabezado TCP, pero siendo administradores de sistemas, ése es otro lio que podemos pasar por alto).

Finalmente, la última pieza del encabezado TCP es el *apuntador de urgente*. Este valor se observa cuando se fija la bandera URG y se le dice a la pila TCP receptora que algunos datos muy importantes se inician en la desviación señalada por dicho apuntador, en relación con el número de secuencia. Se supone que la pila TCP retransmite esta información a la aplicación, de modo que sabe que debe tratar esos datos con importancia especial.

En realidad, el lector se verá presionado a un paquete en el que se usa el bit de URG. Siempre. La mayor parte de las aplicaciones no tienen manera de saber si los datos que les enviaron son urgentes o no y, en realidad, a la mayor parte de las aplicaciones no les importa. Como resultado, debe golpearle a usted una pequeña fibra de paranoia si en verdad ve las banderas de urgente en su red. Asegúrese que no es parte de una sonda del exterior que intenta explotar los errores que haya en su pila TCP y causar que se caiga su servidor (no se preocupe respecto de Linux; sabe cómo manejar con corrección el bit de urgente).

UDP

En comparación con los encabezados TCP, las UDP son mucho más sencillas. Empecemos con echar una mirada a la figura 11-7.

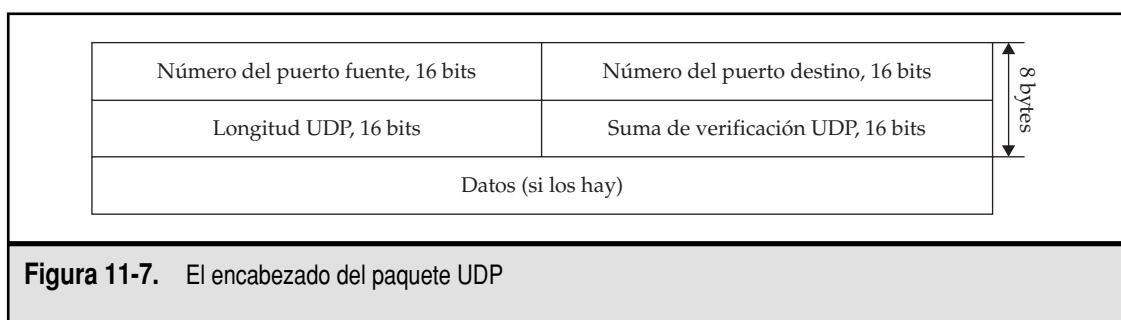
En los primeros campos del encabezado UDP se encuentran los puertos fuente y destino. Desde el punto de vista conceptual, éstos son los mismos que los números de puerto TCP. En la salida de **tcpdump** aparecen de manera muy semejante. Como ejemplo, miremos una consulta DNS para resolver www.djdan.com en una dirección IP, con el comando **tcpdump -n -t port 53**:

```
192.168.1.1.1096 > 192.168.1.8.53: 25851+ A? www.djdan.com. (31)
```

En esta salida podemos ver que el puerto fuente de este paquete UDP es el 1096 y el de destino es el 53. El resto de la línea es la consulta DNS dividida en una forma legible para las personas.

El campo que sigue en el encabezado UDP es la longitud del paquete. **tcpdump** no presenta esta información.

Finalmente, el último campo es la suma de verificación UDP. Ésta la usa UDP para validar que los datos han llegado a su destino sin corrupción. Si la suma de verificación está corrupta, **tcpdump** se lo dirá.



UNA CONEXIÓN TCP COMPLETA

Como discutimos con anterioridad, TCP soporta el concepto de *conexión*. Para quedar establecida, cada conexión debe pasar por una secuencia de pasos; una vez que ambos lados han terminado de enviarse datos, deben recorrer otra secuencia para cerrar la conexión.

En esta sección repasamos el proceso completo de una simple solicitud HTTP y vemos el proceso como lo ve **tcpdump**. Note que todos los registros de **tcpdump** de esta sección se generaron con el comando **tcpdump -n -t port 80**. Por desgracia, debido a la naturaleza compleja de TCP, no podemos cubrir todos los escenarios posibles que puede tomar una conexión TCP. Sin embargo, la cobertura que se da aquí debe ser suficiente para ayudarle a determinar cuando las cosas van mal en la red y no en el servidor.

Apertura de una conexión

Para cada conexión que abre, TCP pasa por un *apretón de manos de tres vías*. La razón de esto es que permite a los dos lados enviarse entre sí su información de estado y darse una posibilidad de reconocer la recepción de los datos.

El primer paquete lo envía el anfitrión que desea abrir la conexión con un servidor. Para esta discusión, llamaremos a este anfitrión como el *cliente*. El cliente envía un paquete TCP sobre IP y fija la bandera TCP en SYN. El número de secuencia es el inicial que el cliente usará para todos los datos que enviará al otro anfitrión (al cual le daremos el nombre de *servidor*).

El segundo paquete lo envía el servidor al cliente. Este paquete contiene dos banderas TCP fijas: SYN y ACK. La finalidad de la ACK es decirle al cliente que ha recibido el primer paquete (SYN). Esto se verifica dos veces al colocar el número de secuencia del cliente en el campo de reconocimiento. La finalidad de la SYN es decirle al cliente con cuál número de secuencia el servidor estará enviando sus respuestas.

Por último, el tercer paquete va del cliente al servidor. Sólo tiene fijado el bit ACK en las banderas TCP con el fin de dar el reconocimiento al servidor de que recibió su SYN. Este paquete ACK tiene el número de secuencia del cliente que está en el campo de número de secuencia y el número de secuencia del servidor que está en el campo de reconocimiento.

¿Suena un poco confuso de nuevo? No se preocupe; así es. Tratemos de aclararlo con un ejemplo real de **tcpdump**. El primer paquete lo envía 192.168.1.1 a 207.126.116.254 y se mira como esto (note que, en realidad, las dos líneas son una línea muy larga):

```
192.168.1.1.1367 > 207.126.116.254.80: S 2524389053:2524389053 (0)
win 32120 <mss 1460,sackOK,timestamp 26292983 0,nop,wscale 0> (DF)
```

Podemos ver que el número de puerto del cliente es el 1367 y el del servidor es el 80. La **S** significa que está establecido el bit SYN y que el número de secuencia es el 2524389053. El 0 que está entre paréntesis después del número de secuencia quiere decir que no se tienen datos en este paquete. Después de que se especifica que la ventana tiene un tamaño de 32 120 bytes, vemos que **tcpdump** nos ha mostrado cuáles opciones TCP fueron parte del paquete. Como administrador de sistema, la única opción que vale la pena hacer notar es el valor MSS (Maximum Segment Size). Este valor nos dice el tamaño máximo al que TCP está detectando para un paquete no seg-

mentado, para esa conexión dada. Las conexiones que requieren valores MSS pequeños, debido a las redes que se les está haciendo recorrer, por lo general requieren más paquetes para transmitir la misma cantidad de datos. Más paquetes significan más carga general y eso significa que se requiere más CPU para procesar una conexión dada.

Advierta que no se tiene fijado bit de reconocimiento y que no existe campo de reconocimiento para imprimir. ¡Esto se debe a que el cliente todavía no cuenta con número de secuencia por reconocer! Momento para el segundo paquete del servidor al cliente:

```
207.126.116.254.80 > 192.168.1.1.1367: S 1998624975:1998624975 (0)
ack 2524389054 win 32736 <mss 1460>
```

Como el primer paquete, el segundo tiene fijado el bit SYN, lo que significa que le está diciendo al cliente con cuál empezará su número de secuencia (en este caso, 1998624975). Está bien que el cliente y el servidor usen números diferentes de secuencia. Aunque lo que es importante es que el servidor reconoce la recepción del primer paquete del cliente al activar el bit ACK y fijar el campo de reconocimiento en 2524389054 (el número de secuencia que el cliente usó para enviar el primer paquete más uno).

Ahora que el servidor ha reconocido la recepción del SYN del cliente, este último necesita reconocer la recepción del SYN del servidor. Esto se hace con un tercer paquete que sólo tiene fijado el bit ACK en sus banderas TCP. Este paquete se mira como esto:

```
192.168.1.1.1367 > 207.126.116.254.80: . 1:1(0) ack 1 win 32120 (DF)
```

Podemos ver con claridad que sólo tiene fijado un bit TCP: ACK. El valor del campo de reconocimiento se muestra como 1. Pero, ¡un momento! ¿No debería estar reconociendo 1998624975? Bien, no se preocupe; así es. **tcpdump** ha sido suficientemente hábil para cambiar en forma automática hacia un modo en el que se imprime la secuencia relativa y los números de reconocimiento, en lugar de los números absolutos. Esto hace que la salida sea mucho más fácil de leer. De modo que, en este paquete, el valor de reconocimiento de 1 quiere decir que está reconociendo el número de secuencia del servidor más uno.

Ahora tenemos una conexión por completo establecida.

Por tanto, ¿por qué todo el barullo para empezar una conexión? ¿Por qué el cliente no puede enviar un solo paquete al servidor diciendo “Quiero empezar a hablar, ¿está bien?” y hacer que el servidor envíe de regreso un “¿está bien?” La razón es que sin los tres paquetes yendo de uno al otro lado, ninguno de los dos está seguro de que el otro lado ha recibido el primer paquete SYN, y ese paquete es crucial para la capacidad de TCP de proporcionar un transporte confiable y en orden.

Transferencia de datos

Con una conexión plenamente establecida en su lugar, los dos lados pueden enviar datos. Puesto que estamos usando como ejemplo una solicitud HTTP, primero veremos al cliente generar una simple petición de una página Web. La salida **tcpdump** se mira como esto:

```
192.168.1.1.1367 > 207.126.116.254.80: P 1:8(7) ack 1 win 32120 (DF)
```

Aquí vemos al cliente enviar 7 bytes al servidor con el bit PSH fijado. La intención del bit PSH es decirle al receptor que procese de inmediato los datos, pero debido a la naturaleza de la interfaz de redes de Linux para las aplicaciones (enchufes), es innecesario fijar el bit PSH. Linux (como todos los sistemas operativos basados en enchufes) procesa en forma automática los datos y los pone a disposición de la aplicación para que los lea tan pronto como pueda.

Junto con el bit PSH está el bit ACK. Esto se debe a que TCP siempre envía el bit ACK en los paquetes salientes. El valor del reconocimiento está fijado en 1, lo cual, con base en la estructura de conexión que observamos en la sección anterior, quiere decir que no ha habido datos nuevos que sea necesario reconocer.

Dado que ésta es una transferencia HTTP, resulta seguro suponer que, como es el primer paquete que va del cliente al servidor, es probable que sea la propia solicitud.

Ahora el servidor envía una respuesta al cliente con este paquete:

```
207.126.116.254.80 > 192.168.1.1.1367: P 1:767(766) ack 8 win 32736 (DF)
```

Aquí el servidor está enviando 766 bytes al cliente y reconociendo los primeros 8 bytes que éste le envió. Ésta probablemente es la respuesta HTTP. Puesto que sabemos que la página que solicitó es muy pequeña, es probable que éstos sean todos los datos que se van a enviar en esta solicitud.

El cliente reconoce estos datos con el paquete siguiente:

```
192.168.1.1.1367 > 207.126.116.254.80: . 8:8(0) ack 767 win 31354 (DF)
```

Éste es un *reconocimiento puro*, lo que quiere decir que el cliente no envió datos, pero sí reconoció hasta el byte 767 que el servidor envió.

El proceso del servidor enviando algunos datos y, a continuación, obteniendo un reconocimiento por parte del cliente puede continuar mientras haya datos que sea necesario enviar.

Cierre de la conexión

Las conexiones TCP tienen la opción de finalizar sin elegancia. Esto equivale a decir que uno de los lados le puede decir al otro “¡detente *ahora!*” Las suspensiones sin elegancia se realizan con la bandera RST (reset), la cual el receptor no reconoce después de recibirla. Esto es para evitar que los dos anfitriones entablen una “guerra de RST”, en donde uno de los lados reinicializa y el otro responde con una reinicialización, causando de este modo un efecto de ping-pong inacabable.

Empecemos con el examen de una suspensión limpia de la conexión HTTP que hemos estado observando hasta ahora. En el primer paso de la suspensión de una conexión, el lado que está listo para cerrarla envía un paquete con el bit FIN fijado, con lo que indica que ha finalizado. Una vez que el anfitrión ha enviado un paquete FIN para una conexión en particular, no se le permite enviar otra cosa que no sean reconocimientos. Esto también significa que aun cuando puede haber finalizado, el otro lado todavía puede enviarle datos. No es sino hasta que ambos lados envían un FIN que se considera que ambos han finalizado. Y, como el paquete SYN, el FIN debe recibir un reconocimiento.

En los dos paquetes siguientes vemos que el servidor le dice al cliente que ha finalizado de enviar datos y el cliente reconoce esto:

```
207.126.116.254.80 > 192.168.1.1.1367: F 767:767(0) ack 8 win 32736
```

```
192.168.1.1.1367 > 207.126.116.254.80: . 8:8(0) ack 768 win 31353 (DF)
```

Entonces vemos lo que sucede en sentido opuesto. El cliente envía un FIN al servidor y esto lo reconoce:

```
192.168.1.1.1367 > 207.126.116.254.80: F 8:8(0) ack 768 win 32120 (DF)
207.126.116.254.80 > 192.168.1.1.1367: . 768:768(0) ack 9 win 32735 (DF)
```

Y eso es todo lo que hay que hacer para realizar una suspensión elegante de la conexión.

Como indicamos con anterioridad, una suspensión sin elegancia es sencillamente que uno de los lados le envíe al otro el paquete RST, lo cual se mira como esto:

```
192.168.1.1.1368 > 207.126.116.254.80: R 93949335:93949349(14) win 0
```

En este ejemplo, 192.168.1.1 está finalizando una conexión con 207.126.116.254 al enviar una re-inicialización. Después de recibir este paquete, la ejecución de **netstat** en el 207.126.116.254 (que sucede que es otro servidor Linux) afirmó que la conexión está por completo cerrada.

CÓMO FUNCIONA ARP

El Address Resolution Protocol (ARP, Protocolo de resolución de direcciones) es un mecanismo que permite a IP convertir las direcciones Ethernet en direcciones IP. Esto es importante porque cuando envía un paquete por una red Ethernet, es necesario poner la dirección Ethernet del anfitrión destino.

La razón para que sepáremos ARP de Ethernet, IP, TCP y UDP es que los paquetes ARP no van hacia arriba de la trayectoria del paquete normal. En lugar de ello, debido a que ARP tiene su propio tipo de encabezado Ethernet (0806), el controlador de Ethernet envía el paquete al subsistema manejador de ARP, el cual nada tiene que ver con TCP/IP.

Los pasos básicos de ARP son los siguientes:

1. El cliente mira en su caché ARP para ver si tiene una conversión entre su dirección IP y su dirección Ethernet (el lector puede ver su caché ARP al ejecutar **arp -a** en su sistema).
2. Si no se encuentra una dirección Ethernet para la dirección IP solicitada, se envía hacia fuera un paquete de difusión pidiendo una respuesta de la persona con la IP que queremos.
3. Si el anfitrión con esa dirección IP está en la LAN, responderá a la petición ARP, informando de este modo al remitente de cuál es su combinación dirección Ethernet/dirección IP.
4. El cliente guarda esta información en su caché y ahora está listo para estructurar un paquete para la transmisión.

Podemos ver un ejemplo de esto, a partir de **tcpdump**, con el comando **tcpdump -e -t -n arp**:

```
0:a0:cc:56:fc:e4 0:0:0:0:0:0 arp 60: arp who-has 192.168.1.1 tell 192.168.1.8
0:10:4b:cb:15:9f 0:a0:cc:56:fc:e4 arp 42: arp reply 192.168.1.1 (0:10:4b:cb:15:9f) is-at
0:10:4b:cb:15:9f
```

El primer paquete es uno de difusión en el que se pide a todos los anfitriones de la LAN la dirección Ethernet para 192.168.1.1. El segundo paquete es una respuesta de 192.168.1.1 dando su conversión de dirección IP/MAC.

Por supuesto, esto conduce a la pregunta: "Si podemos hallar la dirección MAC del anfitrión destino usando una difusión, ¿por qué no sencillamente enviar todos los paquetes por la difusión?" La respuesta tiene dos partes: la primera es que el paquete de difusión requiere que los anfitriones de la LAN que reciban ese paquete se tomen un momento y lo procesen. Esto significa que si dos anfitriones están teniendo una conversación muy intensa (como una transferencia de archivos grandes), todos los demás anfitriones de la misma LAN incurirían en una gran cantidad de carga general en la verificación de paquetes que no les pertenecen. La segunda razón es que el hardware de operación en red (como los comutadores) se apoya en direcciones Ethernet para reenviar con rapidez los paquetes hacia el lugar correcto y minimizar la congestión de la red. En cualquier momento en que un comutador ve un paquete de difusión, debe reenviar ese paquete a *todos* sus puertos. Esto hace que un comutador no sea mejor que un nodo central.

"Ahora bien, si necesitamos la dirección MAC del anfitrión destino para enviarle un paquete, ¿eso no significa que tenga que enviar una petición ARP a los anfitriones que están situados de uno a otro lado de la Internet?" La respuesta es un tranquilizador *no*.

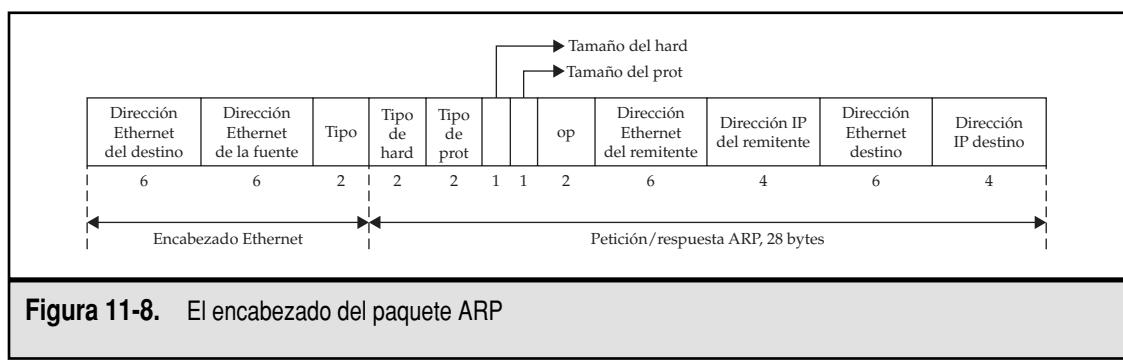
Cuando IP entiende hacia dónde debe encaminar un paquete, primero verifica la tabla de encaminamiento. Si puede hallar la entrada apropiada de la ruta, IP busca una *ruta predeterminada*. Éste es el camino que debe tomarse, cuando todo lo demás falla. Lo normal es que la ruta predeterminada apunte hacia un router o firewall que entienda cómo reenviar los paquetes hacia el resto del mundo.

Esto quiere decir que, cuando un anfitrión necesita enviar algo hacia otro servidor de uno a otro lado de Internet, sólo necesita saber cómo hacer llegar el paquete al router y, por lo tanto, sólo necesita conocer la dirección MAC del mismo.

Para ver que esto sucede en su red, realice un **tcpdump** en su anfitrión y, a continuación, visite un sitio Web que esté en cualquier parte de Internet, como www.yahoo.com. Verá una petición ARP de su máquina para su ruta predeterminada, una respuesta de ésta y, a continuación, el primer paquete de su anfitrión con la IP destino del servidor Web remoto.

El encabezado ARP: ¡ARP también funciona con otros protocolos!

El protocolo ARP no es específico para Ethernet e IP. Para ver por qué, echemos un vistazo al encabezado ARP (vea la figura 11-8).



El primer campo que vemos en el encabezado de ARP es el tipo de hard. En este campo se especifica el tipo de dirección del hardware (Ethernet tiene el valor de 1).

El campo que sigue es el tipo de prot. En éste se especifica la dirección del protocolo que se está convirtiendo. En el caso de IP, ésta se especifica en 0800 (hexadecimal).

En los campos tamaño de hard y tamaño de prot que están inmediatamente después le dicen a ARP lo grandes que son las direcciones que está convirtiendo. Ethernet tiene un tamaño de 6 y el de IP es de 4.

El campo op le dice a ARP qué se necesita hacer. Las peticiones de ARP son 1 y las respuestas de éste son 2.

NOTA Existe una variante de ARP, llamada RARP (lo cual significa Reverse ARP, ARP a la inversa). RARP tiene valores diferentes para el campo op.

Por último, se tienen los campos que estamos tratando de convertir. Una petición tiene las direcciones Ethernet e IP, así como la dirección IP destino llenas. La respuesta llena la dirección Ethernet destino y responde al remitente.

UNIÓN DE REDES IP

Ahora que tenemos algunos de los fundamentos de TCP/IP debajo de nuestro cinturón, lancemos una mirada sobre la manera como funcionan y peguemos las redes. En esta sección cubrimos las diferencias entre anfitriones y redes, netmasks, encaminamiento estático y algunos aspectos básicos en el encaminamiento dinámico.

La finalidad de esta sección no es mostrarle cómo configurar un router Linux, sino presentar los conceptos. Aun cuando puede hallarlo menos excitante que jugar en realidad, encontrará que comprender los aspectos básicos hacen el juego un poco más interesante. Lo que es más importante, si el lector estuviera considerando solicitar un trabajo de administrador de un sistema Linux, éstas podrían ser las cosas que saltarían como parte de las preguntas de la entrevista. Bien, si alguna vez yo tuviera que entrevistarla, ¡puede esperarlo!

Anfitriones y redes

La Internet es un grupo grande de redes interconectadas. Todas estas redes han convenido en conectarse con alguna otra red, con lo que se logra que todos se conecten entre sí. A cada una de estas redes componentes se les asigna una dirección de red.

De manera tradicional, en una dirección IP de 32 bits, por lo común la componente de la red consume hasta 8, 16 o 24 bits para codificar una red clase A, B o C, respectivamente. Como el resto de los bits de la dirección IP se usan para enumerar el anfitrión que se encuentra dentro de la red, entre menos bits se usen para describir la red, se dispondrá de más bits para enumerar los anfitriones. Por ejemplo, la redes clase A han dejado 24 bits para la componente de los anfitriones, lo cual quiere decir que puede haber arriba de 16 777 214 anfitriones dentro de la red (las clases B y C tienen 65 534 y 254 nodos, respectivamente).

NOTA También existen los rangos clases D y E. La clase D se usa para emisión hacia múltiples receptores y la E se reserva para uso experimental.

Con el fin de organizar mejor las diversas clases de redes, desde el principio de la vida de IP se estableció que unos cuantos de los primeros bits decidirían a cuál clase pertenecía la red. En beneficio de la facilidad de lectura, el primer *octeto* de la dirección IP especifica la clase.

NOTA Un octeto es 8 bits, lo cual en la notación decimal típica con puntos equivale al número antes de un punto. Por ejemplo, en la dirección IP 192.168.1.42, el primer octeto es 192, el segundo es 168, y así sucesivamente.

Los rangos son como sigue:

Clase	Rango del octeto
A	0–126
B	128–192.167
C	192.169–223

Es probable que el lector advierta algunos huecos en los rangos. Esto se debe a que existen algunas direcciones especiales que se reservan para usos especiales. La primera dirección especial es una con la que probablemente el lector esté familiarizado: 127.0.0.1. Ésta también se conoce como la *dirección de bucle cerrado*. Se establece en cada anfitrión que use IP, de modo que pueda referirse a sí mismo. Parece un poco raro hacerlo de esta manera, pero sólo porque un sistema es capaz de hablar IP no significa que tiene una dirección IP asignada a él! Por otra parte, la dirección 127.0.0.1 está virtualmente garantizada (si no está allí, es más que probable que algo ha ido mal).

Otros tres rangos son notables: todo IP en la red 10.0.0.0, las redes 172.16-172.31 y la red 192.168 se considera un *IP privado*. No se permite que estos rangos se asigne a nadie en la Internet y, por lo tanto, usted puede usarlos en sus redes internas.

NOTA Definimos las redes internas como aquellas que se encuentran detrás de un firewall, no conectadas en realidad a la Internet, o que tienen un router realizando traducción de direcciones de la red en el borde de la red que se conecta a la Internet (la mayor parte de los firewalls también realizan esta traducción de direcciones).

Creación de subredes

Imagine una red con unos cuantos miles de anfitriones en ella, lo cual no es irrazonable en una compañía de tamaño medio. Tratar de ligarlos a todos en una sola red grande es probable que conduciría a que usted se tirara de los cabellos, hiciera chocar su cabeza contra una pared o, quizás, las dos cosas. Y eso sólo es el aspecto imaginario.

Las razones para no mantener una red como una sola entidad grande van desde aspectos muy técnicos hasta muy políticos. En el frente técnico existen limitaciones para todas las tecnologías acerca de cuán grande puede llegar a ser una red, antes de que se haga demasiado grande. Por ejemplo, Ethernet no puede tener más de 1 024 anfitriones en un solo dominio de conflictos.

Siendo realista, tener más de una docena en una red incluso poco ocupada causará graves problemas de rendimiento. Incluso los anfitriones emigrantes hacia los conmutadores no resuelven el problema por completo, ya que también los conmutadores tienen limitaciones respecto a cuántos anfitriones pueden atender.

Por supuesto, es probable que usted tenga que manejar aspectos de administración antes de chocar contra las limitaciones de los conmutadores; administrar una sola red grande es muy difícil. Además, a medida que crece una organización, cada uno de los departamentos se va dividiendo en compartimientos. Recursos Humanos suele ser el primer candidato que necesita una red segura por sí mismo, de manera que ingenieros entrometidos no estén atisbando en cosas que no deben. Con el fin de soportar una necesidad como esa, usted debe *crear subredes*.

Suponiendo que nuestra red corporativa sea de 10.0.0.0, podríamos crear las subredes estructurando redes clase C dentro de ella, como 10.1.1.0, 10.1.2.0, 10.1.3.0, etcétera. Estas redes más pequeñas tendrían componentes de red de 24 bits y componentes de anfitrión de 8 bits. Puesto que los primeros 8 bits se usarían para identificar nuestra red corporativa, podríamos usar los 16 bits restantes de la componente de red para especificar la subred, lo que nos da 65 534 subredes posibles. Por supuesto, ¡no tiene que usar todas ellas!

NOTA Como hemos visto con anterioridad en este capítulo, por lo general las direcciones de redes tienen fijada en ceros la componente del anfitrión de una dirección IP. Esta convención facilita que otros humanos reconozcan cuáles direcciones corresponden a las redes completas y cuáles corresponden específicamente a anfitriones.

Netmasks

La finalidad de una *netmask* es decirle a la pila IP cuál parte de la dirección IP es la red y cuál es el anfitrión. Esto le permite a la pila determinar si una dirección IP de destino está en la LAN o si necesita enviarla a un router para que la reenvíe a alguna otra parte.

La mejor manera de empezar a ver las netmasks es mirar las direcciones IP y las netmasks en sus representaciones binarias. Miremos la dirección 192.168.1.42 con la netmask 255.255.255.0:

Decimal con puntos	Binario
192.168.1.42	11000000 10101000 00000001 00101010
255.255.255.0	11111111 11111111 11111111 00000000

En este ejemplo, queremos averiguar qué parte de la dirección IP 192.168.1.42 es red y qué parte es anfitrión. Ahora bien, según la definición de netmask, aquellos bits que son cero son parte del anfitrión. Dada esta definición, vemos que los tres primeros octetos forman la dirección de la red y el último forma el anfitrión.

Al discutir las direcciones de redes con otras personas, a menudo resulta práctico poder decir la dirección de la red sin tener que dar la dirección IP original y la netmask. Por fortuna, esta dirección de la red se puede calcular, dada la dirección de la red y la netmask, con la aplicación de una operación AND en términos de bits.

La manera en que funciona la operación AND en términos de bits se puede explicar mejor si se observa el comportamiento de dos bits a los que se les está aplicando AND. Si los dos bits son 1, entonces el resultado de la AND también es 1. Si cualquiera de los dos bits es cero (o ambos lo son), el resultado es 0. Podemos ver esto con mayor claridad en esta tabla:

Bit 1	Bit 2	Resultado de AND en bits
0	0	0
0	1	0
1	0	0
1	1	1

De modo que si se calcula la operación AND en términos de bits aplicada a 192.168.1.42 y 255.255.255.0 da el patrón de bits 11000000 10101000 00000001 00000000. Advierta que los tres primeros octetos permanecen idénticos y que el último se convierte en puros ceros. En la notación decimal, esto se lee 192.168.1.0.

NOTA Recuerde que necesitamos entregar una dirección IP a la dirección de la red y una IP a la dirección de difusión. En este ejemplo, la dirección de la red es 192.168.1.0 y la de difusión es 192.168.1.255.

Recorramos otro ejemplo. En esta ocasión, queremos hallar el rango de direcciones de las que disponemos para la dirección de red 192.168.1.176 con una netmask de 255.255.255.240 (es común que los ISP den este tipo de netmask a los negocios DSL y a los clientes T1).

Una división del último octeto de la netmask hace ver que el patrón de bits para 240 es 11110000. Esto significa que los tres primeros octetos de la dirección de la red más cuatro bits en el cuarto octeto se mantienen constantes (255.255.255.240 en binario es 11111111 11111111 11111111 11110000). Como los últimos cuatro bits son variables, sabemos que contamos con 16 direcciones posibles ($2^4 = 16$). Por tanto, nuestro rango va desde 192.168.1.176 hasta 192.168.1.192 (192 – 176 = 16).

Debido a que es tan tedioso teclear las netmasks completas, mucha gente ha empezado a usar un formato abreviado en donde la dirección de la red va seguida de una barra diagonal y el número de bits de la netmask. De modo que la dirección de la red 192.168.1.0 con una netmask de 255.255.255.0 se abreviaría como 192.168.1.0/24.

NOTA El proceso de usar netmasks que no caen en las fronteras de la clase A, B o C también se conoce como encaminamiento interdominios sin clase (CIDR). El lector puede ver más acerca de CIDR en RFC 1817 (<http://www.rfc-editor.org/rfc/rfc1817.txt>).

Encaminamiento estático

Cuando se quieren comunicar dos anfitriones que se encuentran en la misma LAN, es bastante fácil para ellos hallarse entre sí. Sencillamente difunden un mensaje ARP, obtienen la dirección MAC del otro anfitrón y usted se marcha. Pero cuando el segundo anfitrón no es local, las cosas se vuelven más difíciles.

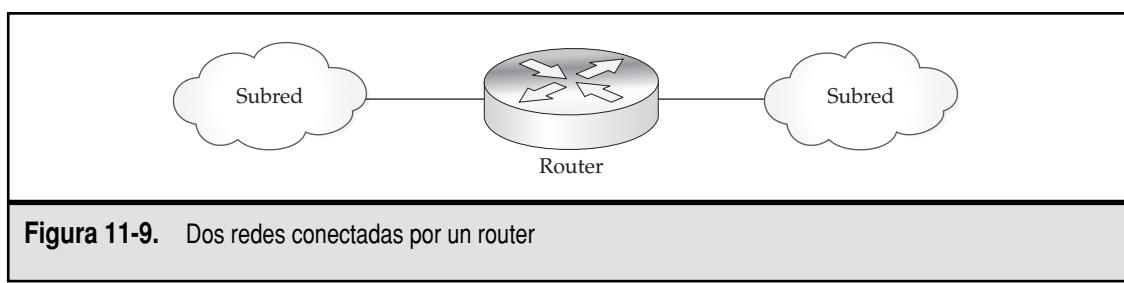
Para que dos o más LAN se comuniquen entre sí, se necesita colocar un router. La finalidad de éste es conocer acerca de la topología de redes múltiples. Cuando usted quiere comunicarse con otra red, su máquina fijará la IP de destino como la del anfitrión en la otra red, pero la dirección MAC de destino será para el router. Esto permite que este último reciba el paquete y examine la IP destino, y como sabe que la IP está en la otra red, reenviará el paquete. Lo opuesto también se cumple para los paquetes que están viniendo de la otra red hacia la mía (vea la figura 11-9).

A su vez, el router debe saber cuáles redes están enchufadas en él. Esta información se conoce como *tabla de encaminamiento*. Cuando el router se informa de modo manual acerca de cuáles trayectorias puede tomar, la tabla se conoce como *estática* y, de allí, el término *encaminamiento estático*. Una vez que una persona introduce las rutas en la tabla, no se pueden cambiar, hasta que un operador humano regrese a cambiarlas.

Por desgracia, los routers son dispositivos un tanto caros. Por lo general, son piezas dedicadas del hardware que optimizan con intensidad con el fin de reenviar paquetes de una interfaz a otra. Por supuesto, usted puede hacer un router basado en Linux (discutimos esto en el capítulo 12) usando una PC en existencia que tenga dos o más tarjetas de redes. Esas configuraciones son rápidas y suficientemente baratas para las redes más pequeñas. De hecho, muchas compañías ya están empezando a hacer esto, ya que las PC más antiguas que son demasiado lentas para ejecutar los navegadores Web y las aplicaciones de procesamiento de textos más recientes todavía son bastante rápidas para realizar el encaminamiento (¡una máquina de clase Pentium de baja versión es suficientemente rápida para mantener una conexión Ethernet de 100Mb completa! Muchos routers dedicados incluso usan hardware más sencillo y más lento).

La desventaja para los routers basados en Linux tiene más que ver con el hardware de la PC que con el propio Linux. Por su naturaleza, las PC se construyen para ser menos robustas que el hardware dedicado de la red. Además, tener un sistema operativo que permita a los usuarios conectarse para hacer cosas diferentes a las de encaminamiento significa que usted corra el riesgo de que administradores que no son principales agreguen carga a un router que no deben estar tocando (este problema es independiente del sistema operativo de usted. ¡Observe cómo el rendimiento de cualquier servidor o router cae por debajo del piso cuando alguien piensa que un protector de pantalla basado en OpenGL sería una linda adición!).

Como con cualquier dispositivo, tómelo dentro del contexto de sus necesidades, presupuesto y habilidades. La fuente abierta y Linux son grandes herramientas, pero como cualquier otro tema, asegúrese de que está usando la herramienta correcta para el trabajo.



Tablas de encaminamiento

Como se mencionó con anterioridad, las tablas de encaminamiento son listas de direcciones de redes, netmasks e interfaces destino. Una versión simplificada de una tabla se podría ver como esto:

Dirección de la red	Netmask	Interfaz destino
192.168.1.0	255.255.255.0	Interfaz 1
192.168.2.0	255.255.255.0	Interfaz 2
192.168.3.0	255.255.255.0	Interfaz 3
Predeterminada	0.0.0.0	Interfaz 4

Cuando un paquete llega a un router que tiene una tabla de encaminamiento como ésta, recorrerá la lista de las rutas y aplicará cada netmask a la dirección IP destino. Si la dirección resultante de la red es igual a la dirección de la red que está en la tabla, el router sabe dirigir el paquete hacia esa interfaz.

De modo que, por ejemplo, el router recibe un paquete con la dirección IP de destino fijada como 192.168.2.233. La primera entrada de la tabla tiene la netmask 255.255.255.0. Cuando esta netmask se aplica a 192.168.2.233, el resultado no es 192.168.1.0, de modo que el router se mueve hacia la segunda entrada. Como la primera entrada de la tabla, esta ruta tiene la netmask de 255.255.255.0. El router aplicará esto a 192.168.2.233 y encontrará que la dirección resultado de la red es igual a 192.168.2.0. De modo que ahora se encuentra la ruta apropiada. El paquete se reenvía hacia la interfaz 2.

Si llega un paquete que no corresponde a las tres primeras rutas, hará coincidir el caso predeterminado. En nuestra tabla de encaminamiento, esto hará que el paquete se reenvíe hacia la interfaz 4. Lo más probable es que ésta sea una compuerta hacia la Internet.

Limitaciones del encaminamiento estático

El ejemplo de encaminamiento estático que hemos usado es típico de las redes más pequeñas. Sólo existe un manojo de redes que necesitan comunicarse entre sí y no se van a cambiar con frecuencia.

Sin embargo, existen limitaciones para esta técnica. La limitación más grande es humana: usted es el responsable de actualizar todos sus routers con información nueva, siempre que haga cualesquiera cambios. Aun cuando esto suele ser muy fácil de hacer en una red pequeña, quiere decir que hay lugar para que se cometan errores. Además, a medida que su red crece y se agregan más rutas, es más probable que la tabla de encaminamiento se vuelva más difícil de manejar de esta manera.

La segunda limitación, pero casi tan significativa, es que el tiempo que tarda el router para procesar un paquete puede aumentar según el número de rutas que haya. Con sólo tres o cuatro rutas, esto no es un gran problema; pero conforme usted empieza a tener docenas de rutas, la carga general se puede volver notable.

Dadas estas dos limitaciones, lo mejor es usar las rutas estáticas sólo en redes pequeñas.

Encaminamiento dinámico con RIP

A medida que crecen las redes, aumenta la necesidad de dividirlas en subredes. Llegará un momento en que usted encontrará que cuenta con una gran cantidad de subredes a las que no puede

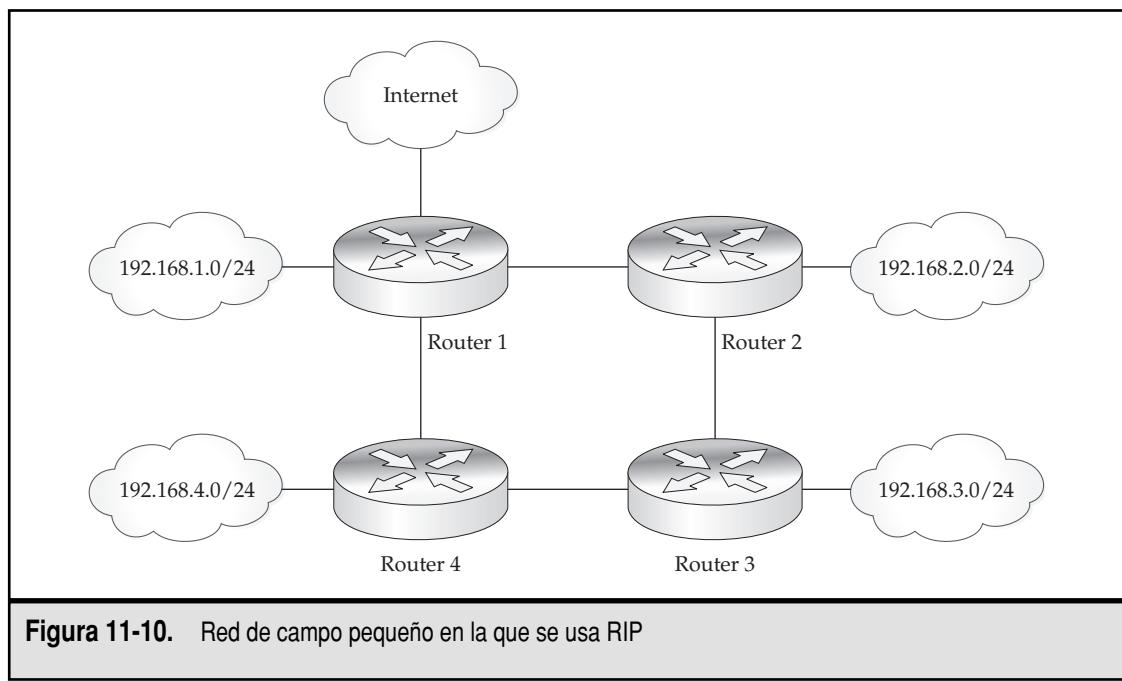
seguir la ruta a todas con facilidad, en especial si están siendo administradas por administradores diferentes. Por ejemplo, una subred puede necesitar dividir su red a la mitad por razones de seguridad. En una situación así de compleja, ir de un lado a otro para decirle a cada uno que actualice sus tablas de encaminamiento sería una pesadilla real y conduciría a toda clase de dolores de cabeza debidos a la red.

La solución para este problema es usar el *encaminamiento dinámico*. La idea que se encuentra detrás de éste es que cada router sólo conoce las redes inmediatamente adyacentes cuando arranca. Entonces anuncia a los demás routers conectados a él lo que conoce y esos otros le responden con lo que ellos conocen. Piense en esto como una publicidad “comunicada verbalmente” por su red. Usted le dice a las personas que lo rodean acerca de su red, entonces ellos les dicen a sus amigos, y sus amigos les dicen a sus amigos, y así sucesivamente. Llegará el momento en que quienquiera que esté conectado a la red sabe acerca de su nueva red.

Las redes que cubren todo un campo (como una compañía grande con muchos departamentos) normalmente verá este método de anunciar la información de rutas. En la fecha en que esto se está escribiendo, los dos protocolos de encaminamiento de uso más común son RIP y OSPF.

RIP (abreviatura de Routing Information Protocol, Protocolo de encaminamiento de la información) se encuentra en la actualidad en la versión 2. Es un protocolo muy sencillo que es fácil de configurar. Sencillamente déle al router la información acerca de una red (asegurándose de que cada subred de la compañía tenga una conexión con un router que sepa acerca de RIP) y, a continuación, haga que los routers se conecten entre sí. Las emisiones de RIP se producen a intervalos regulares (por lo común, de menos de un minuto) y, en sólo unos cuantos minutos, la red completa del campo sabe acerca de usted.

Veamos la manera en que funcionaría con RIP una red de un campo más pequeño con cuatro subredes. En la figura 11-10 se muestra cómo se conecta la red.



NOTA En beneficio de la sencillez, estamos mencionando en serie los eventos. En la realidad, muchos de estos eventos sucederían en paralelo.

Como se ilustra en esta figura, al router 1 se le hablaría acerca de la 192.168.1.0/24 y acerca de la ruta predeterminada a la Internet. Al router 2 se le hablaría acerca de la 192.168.2.0/24, el router 3 sabría acerca de la 192.168.3.0/24, y así sucesivamente. En el arranque, la tabla de cada router se mira como esto:

Router	Tabla
Router 1	192.168.1.0/24
	Puerta de acceso a Internet
Router 2	192.168.2.0/24
Router 3	192.168.3.0/24
Router 4	192.168.4.0/24

Entonces el router 1 hace una difusión en la que expresa acerca de cuáles sabe. Como los routers 3 y 4 están conectados con él, actualizan sus rutas. Esto hace que la tabla de encaminamiento se mire como esto (las nuevas rutas están en *cursivas*):

Router	Tabla
Router 1	192.168.1.0/24
	Puerta de acceso a Internet
Router 2	192.168.2.0/24
	<i>192.168.1.0/24 a través del router 1</i>
	<i>Puerta de acceso a Internet a través del router 1</i>
Router 3	192.168.3.0/24
Router 4	192.168.4.0/24
	<i>192.168.1.0/24 a través del router 1</i>
	<i>Puerta de acceso a Internet a través del router 1</i>

El router 2 hace su difusión. Los routers 1 y 3 ven estos paquetes y actualizan sus tablas como sigue (las nuevas rutas están en *cursivas*):

Router	Tabla
Router 1	192.168.1.0/24
	Puerta de acceso a Internet
	<i>192.168.2.0/24 a través del router 2</i>

Router 2	192.168.2.0/24 192.168.1.0/24 a través del router 1 Puerta de acceso a Internet a través del router 1
Router 3	192.168.3.0/24 <i>192.168.2.0/24 a través del router 2</i> <i>192.168.1.0/24 a través del router 2</i> <i>Puerta de acceso a Internet a través del router 2</i>
Router 4	192.168.4.0/24 192.168.1.0/24 a través del router 1 Puerta de acceso a Internet a través del router 1

Entonces el router 3 hace su difusión, la cual escuchan los routers 2 y 4. En este punto es en donde las cosas se ponen interesantes, ya que esto introduce información suficiente como para que haya múltiples rutas hacia el mismo destino. Las tablas de encaminamiento ahora se miran como esto (las rutas nuevas están en *cursivas*):

Router	Tabla
Router 1	192.168.1.0/24 Puerta de acceso a Internet
Router 2	192.168.2.0/24 192.168.1.0/24 a través del router 1 Puerta de acceso a Internet a través del router 1 <i>192.168.3.0/24 a través del router 3</i>
Router 3	192.168.3.0/24 192.168.2.0/24 a través del router 2 192.168.1.0/24 a través del router 2 Puerta de acceso a Internet a través del router 2
Router 4	192.168.4.0/24 192.168.1.0/24 a través del router 1 o 3 Puerta de acceso a Internet a través del router 1 o 3 <i>192.168.3.0/24 a través del router 3</i> <i>192.168.2.0/24 a través del router 3</i>

Enseguida, el router 4 hace su difusión. Los routers 1 y 3 escuchan esto y actualizan sus tablas hacia las siguientes (las rutas nuevas están en *cursivas*):

Router	Tabla
Router 1	192.168.1.0/24 Puerta de acceso a Internet <i>192.168.2.0/24 a través del router 2 o 4</i> <i>192.168.3.0/24 a través del router 4</i> <i>192.168.4.0/24 a través del router 4</i>
Router 2	192.168.2.0/24 192.168.1.0/24 a través del router 1 Puerta de acceso a Internet a través del router 1 192.168.3.0/24 a través del router 3
Router 3	192.168.3.0/24 192.168.2.0/24 a través del router 2 192.168.1.0/24 a través del router 2 o 4 Puerta de acceso a Internet a través del router 2 o 4 <i>192.168.4.0/24 a través del router 4</i>
Router 4	192.168.4.0/24 192.168.1.0/24 a través del router 1 Puerta de acceso a Internet a través del router 1 192.168.3.0/24 a través del router 3 192.168.2.0/24 a través del router 3

Una vez que todos los routers pasan por otra ronda de difusiones, la tabla completa se miraría como ésta:

Router	Tabla
Router 1	192.168.1.0/24 Puerta de acceso a Internet 192.168.2.0/24 a través del router 2 o 4 192.168.3.0/24 a través del router 4 o 2 192.168.4.0/24 a través del router 4 o 2
Router 2	192.168.2.0/24 192.168.1.0/24 a través del router 1 o 3 Puerta de acceso a Internet a través del router 1 o 3 192.168.3.0/24 a través del router 3 o 1

Router 3	192.168.3.0/24
	192.168.2.0/24 a través del router 2 o 4
	192.168.1.0/24 a través del router 2 o 4
	Puerta de acceso a Internet a través del router 2 o 4
	192.168.4.0/24 a través del router 4 o 2
Router 4	192.168.4.0/24
	192.168.1.0/24 a través del router 1 o 3
	Puerta de acceso a Internet a través del router 1 o 3
	192.168.3.0/24 a través del router 3 o 1
	192.168.2.0/24 a través del router 3 o 1

¿Por qué es tan importante esta malla? Digamos que el router 2 falla. Si el router 3 se estaba apoyando en el 2 para enviar paquetes a la Internet, de inmediato puede actualizar sus tablas, reflejando que el router 2 ya no está allí y, a continuación, reenvía los paquetes rebotados de la Internet a través del router 4.

El algoritmo de RIP (y por qué debe usar OSPF en lugar de él)

Por desgracia, cuando llega el momento de pensar en la trayectoria más óptima desde una subred hacia otra, RIP no es el protocolo más inteligente. Su método de determinación de cuál ruta tomar se basa en el menor número de routers (saltos) entre él y el destino. Aun cuando esto suena óptimo, lo que este algoritmo no toma en cuenta es cuánto tráfico se encuentra en el enlace o cuán rápido es éste.

Regresando a la figura 11-10, podemos ver en dónde se podría terminar esta situación. Supongamos que el enlace entre los routers 3 y 4 se llega a congestionar mucho. Ahora, si el router 3 quiere enviar un paquete hacia la Internet, RIP todavía evaluará las dos trayectorias posibles (3 a 4 a 1 y 3 a 2 a 1) como si fueran equidistantes. Como resultado, el paquete puede finalizar yendo a través del router 4 cuando es evidente que la trayectoria por el router 2 (cuyos enlaces no están congestionados) sería mucho más rápida.

OSPF (Open Shortest Path First, Abrir primero la trayectoria más corta) es muy semejante a RIP en cuanto a difundir información hacia los otros routers. Lo que lo hace diferente es que, en lugar de seguir con atención cuántos saltos se requieren para ir de un router hacia el otro, se mantiene informado de con cuánta rapidez cada uno de los routers le está hablando a los otros. Como consecuencia, en nuestro ejemplo, en donde el enlace entre los routers 3 y 4 se ha llegado a congestionar, OSPF se dará cuenta de que hay que encaminar un paquete destinado al router 1 a través del 2, y se asegurará de ello.

Otra característica de OSPF es su capacidad para darse cuenta de cuándo una dirección destino tiene dos trayectorias posibles en las que se consumiría una cantidad igual de tiempo. Cuando ve esto, OSPF compartirá el tráfico a través de los dos enlaces, un proceso conocido como *trayectorias múltiples de igual costo*, haciendo de este modo un uso óptimo de los recursos de los que se dispone.

Existen dos “pequeños inconvenientes” con OSPF. El hardware de trabajo en red más antiguo y alguno de calidad más baja pueden no contar con OSPF o tenerlo a un costo sustancialmente más elevado. El segundo inconveniente es la complejidad: RIP es mucho más sencillo de estructurar que OSPF. Para una red pequeña, RIP puede ser, en principio, una buena selección.

ESCUDRÍNO DE TCPDUMP

La herramienta **tcpdump** es en realidad una de las herramientas más poderosas que usted usará como administrador de sistema. La versión GUI de ella, Ethereal (<http://www.ethereal.com>), es incluso una mejor selección cuando se dispone de un sistema de entrada gráfica. Ethereal ofrece todo el poder de **tcpdump** con el beneficio adicional de filtros más ricos, soporte adicional del protocolo, la capacidad de seguir con rapidez las conexiones TCP y algunas estadísticas prácticas.

En esta sección recorreremos algunos ejemplos de la manera en que puede usar **tcpdump**.

Unas cuantas notas generales

Enseguida se dan unas cuantas sugerencias rápidas referentes a estas herramientas, antes que salte hacia ejemplos más avanzados.

Ethereal

Ethereal es una herramienta gráfica para seguir el rastro de paquetes y decodificarlos. Ofrece una gran cantidad de más características que **tcpdump** y es una gran manera de echar una ojeada al interior de varios protocolos. Puede descargar la versión más reciente de Ethereal en <http://www.ethereal.com>.

Una bonita característica adicional de Ethereal es que puede funcionar bajo el entorno nativo de Windows, de modo que si tiene un escritorio de Windows y una gran cantidad de servidores Linux, puede realizar la captura de un paquete en el servidor, retirar la captura y, entonces, verla desde su escritorio, sin necesidad de iniciar una sesión de X Window en un sistema Linux remoto.

Antes de que se excite demasiado acerca de Ethereal, no olvide ensuciarse las manos también con **tcpdump**. En sesiones de detección de fallas no siempre tiene el tiempo o no puede darse el lujo de sacar a la luz Ethereal, y si sólo está buscando una validación rápida de que los paquetes se están moviendo, arrancar una herramienta GUI puede ser un poco más de lo que necesita. La herramienta **tcpdump** ofrece una manera rápida para manejar la situación. Aprendiéndola podrá manejar con rapidez una gran cantidad de situaciones.

SUGERENCIA Sus amigos de Sun Solaris le pueden hablar acerca de **snoop**. Aunque no son idénticas, **tcpdump** y **snoop** tienen muchas similitudes. Si aprende una usted sabrá mucho de la otra.

Lectura y escritura de archivos dump

Si necesita capturar una gran cantidad de datos y guardarlos, querrá usar la opción **-w** con el fin de escribir todos los paquetes en el disco para el procesamiento posterior. Aquí está un ejemplo sencillo:

```
[root@hostA:~]# tcpdump -w /tmp/trace.pcap -i eth0
```

La herramienta **tcpdump** continuará capturando paquetes vistos en la interfaz eth0, hasta que se cierre la terminal, el proceso se anule o se oprima CTRL-C. El archivo resultante lo puede cargar Ethereal o lo puede leer cualquier cantidad de otros programas que puedan procesar capturas formateadas por **tcpdump** (el propio formato del paquete se menciona como "pcap").

NOTA Cuando se usa la opción **-w** con **tcpdump**, no es necesario emitir la opción **-n** con el fin de evitar las búsquedas de DNS para cada dirección IP vista.

Para volver a leer el rastro del paquete, usando **tcpdump**, use la opción **-r**. Al volver a leer el rastro de un paquete, se pueden aplicar filtros y opciones adicionales para influir sobre la manera en que se presentarán los paquetes. Por ejemplo, para mostrar sólo los paquetes ICMP de un archivo de rastros y evitar las búsquedas de DNS conforme se presenta la información, haga lo siguiente:

```
[root@hostA:~]# tcpdump -r /tmp/trace.pcap -n icmp
```

Captura de más por paquete

De modo predeterminado, **tcpdump** se limita a capturar los 68 primeros bytes de un paquete. Si usted sólo está mirando para rastrear algunos flujos y ver lo que está sucediendo en el alambre, esto suele ser suficientemente bueno. Sin embargo, si necesita capturar el paquete entero para la decodificación posterior, necesitará incrementar este valor. Para hacerlo, use la opción **-s** (snaplen). Por ejemplo, para capturar un paquete completo de 1 500 bytes y escribirlo en el disco, podría usar

```
[root@hostA:~]# tcpdump -w /tmp/dump.pcap -i eth0 -s 1500
```

Impacto sobre el rendimiento

Seguir el rastro de un paquete puede tener un impacto sobre el rendimiento, en especial en un servidor intensamente cargado. Existen dos partes relativas al rendimiento: la captura real de los paquetes y la decodificación/impresión de los mismos.

La captura real de los paquetes, aun cuando un tanto costosa, en realidad se puede minimizar con un buen filtro. En general, a menos que la carga de su servidor sea en extremo elevada o esté usted moviendo una gran cantidad de tráfico (una gran cantidad se refiere a cientos de megabits/s), este castigo no es demasiado significativo. El costo que se tiene allí proviene del castigo de mover paquetes del núcleo hacia arriba, hasta la aplicación **tcpdump**, lo cual requiere tanto una copia en la memoria intermedia como un commutador de contexto.

Como comparación, la decodificación/impresión de los paquetes es sustancialmente más cara. La propia decodificación es una fracción pequeña del costo, pero la de la impresión es muy elevada. Si su servidor está cargado, usted quiere evitar la impresión por dos razones: genera carga para formatear las cadenas que constituyen la salida y genera carga para actualizar su pantalla. Este último factor puede ser en especial costoso si está usando una consola en serie, ya que cada byte que se envía por el puerto en serie genera una interrupción de alta prioridad (más alta que las de las tarjetas de redes) que requiere mucho tiempo para procesar debido a que este tipo de puertos son en comparación mucho más lentos que todo lo demás. La impresión de paquetes decodificados por un puerto en serie puede generar suficiente interrupción del tráfico como para hacer que las tarjetas de redes dejen caer los paquetes conforme empiezan a carecer de atención por parte de la CPU principal.

Con el fin de aliviar el esfuerzo del proceso de decodificación/impresión, use la opción **-w** para escribir paquetes en bruto en el disco. El proceso de escribir datos en bruto es mucho más rápido y de costo inferior que el de imprimirlos. Además, escribir paquetes en bruto quiere decir que se brinca el paso completo de decodificación/impresión, ya que eso sólo se hace cuando necesita ver los paquetes.

En pocas palabras, si no está seguro, use la opción **-w** para escribir paquetes en el disco, cópielos en otra máquina y, después, léalos allí.

No capture su propio tráfico de red

Un error común que se comete al usar **tcpdump** es conectarse a través de la red y, a continuación, iniciar una captura. Sin el filtro apropiado, terminará capturando sus paquetes de sesiones, los cuales, a su vez, si los está imprimiendo en la pantalla, puede generar paquetes nuevos, los cuales se capturan, y así sucesivamente. Una manera rápida de pasar por alto su propio tráfico (y el de otros administradores) es sencillamente brincarse el puerto 22 (el puerto ssh) en la captura, de este modo:

```
[root@hostA:~]# tcpdump not tcp port 22
```

Si quiere ver lo que otras personas están haciendo en ese puerto, agregue un filtro que se aplique sólo a su propio anfitrión. Por ejemplo, si usted está vieniendo del 192.168.1.8, puede escribir

```
root@hostA:~]# tcpdump "not (host 192.168.1.8 and tcp port 22)"
```

Note la adición de las comillas. Esto se hizo para no confundir al shell con los paréntesis agregados, los cuales son para **tcpdump**.

Lea la página man

Para ver el poder completo de **tcpdump**, tómese algún tiempo para leer su página man completa. Puede ver la versión más reciente de **tcpdump** en <http://www.tcpdump.org>, y la versión más reciente de la página man http://www.tcpdump.org/tcpdump_man.html.

Uso de **tcpdump** para observar la ruta de un rastro

El programa **traceroute** funciona fijando inicialmente el valor TTL de un paquete UDP en 1 y a continuación enviándolo al anfitrión destino. Esto hace que el primer router a lo largo de la trayectoria deje caer el paquete y envíe de regreso un mensaje ICMP. **traceroute** ve el mensaje ICMP y ahora conoce el primer router en la trayectoria entre el anfitrión fuente y el de destino. Se estructura otro paquete, en esta ocasión con un valor TTL de 2, y se envía. Por supuesto, este paquete hace que se anuncie el segundo router a lo largo de la trayectoria por medio de un mensaje ICMP.

El programa **traceroute** repite este proceso hasta que el propio anfitrión destino genera un mensaje ICMP acerca del paquete que tiene un TTL de 0. Habiendo recibido un mensaje ICMP de cada router entre los dos anfitriones, puede presentar esta lista para usted. Por ejemplo, si queremos ver cada router entre hostA.planetoid.org y 68.121.104.1, veríamos la siguiente salida de **traceroute**:

```
[sshah@hostA:~]$ traceroute -n 68.121.104.1
traceroute to 68.121.104.1 (68.121.104.1), 64 hops max, 44 byte packets
1 68.121.105.174 7.689 ms 8.471 ms 8.556 ms
2 64.164.97.67 9.304 ms 9.494 ms 9.541 ms
3 64.164.97.142 12.532 ms 12.978 ms 12.928 ms
```

Es probable que, en este caso, el punto final sea un router que tiene múltiples direcciones IP asignadas a él; por tanto, el último salto está mostrando una dirección IP diferente que la que elegimos en el comando **traceroute**. De modo que entonces ¿se mira así el rastro real del paquete?

Para hacer que **tcpdump** sea más conciso, agreguemos, la opción **-t**, con lo cual no se imprimirá la hora, y usemos el filtro **udp or icmp**. El comando **tcpdump** usado para capturar esta ruta del rastro es

```
[root@hostA:~]# tcpdump -t -n udp or icmp
68.121.105.170.47762 > 68.121.104.1.33435: udp 16 [ttl 1]
68.121.105.174 > 68.121.105.170: icmp: time exceeded in-transit
68.121.105.170.47762 > 68.121.104.1.33436: udp 16 [ttl 1]
68.121.105.174 > 68.121.105.170: icmp: time exceeded in-transit
68.121.105.170.47762 > 68.121.104.1.33437: udp 16 [ttl 1]
68.121.105.174 > 68.121.105.170: icmp: time exceeded in-transit
68.121.105.170.47762 > 68.121.104.1.33438: udp 16
64.164.97.67 > 68.121.105.170: icmp: time exceeded in-transit [tos 0xc0]
68.121.105.170.47762 > 68.121.104.1.33439: udp 16
64.164.97.67 > 68.121.105.170: icmp: time exceeded in-transit [tos 0xc0]
68.121.105.170.47762 > 68.121.104.1.33440: udp 16
64.164.97.67 > 68.121.105.170: icmp: time exceeded in-transit [tos 0xc0]
68.121.105.170.47762 > 68.121.104.1.33441: udp 16
64.164.97.142 > 68.121.105.170: icmp: 68.121.104.1 udp port 33441 unreachable
68.121.105.170.47762 > 68.121.104.1.33442: udp 16
64.164.97.142 > 68.121.105.170: icmp: 68.121.104.1 udp port 33442 unreachable
68.121.105.170.47762 > 68.121.104.1.33443: udp 16
64.164.97.142 > 68.121.105.170: icmp: 68.121.104.1 udp port 33443 unreachable
```

En el rastro podemos ver al anfitrión A (68.121.105.170) enviando un paquete UDP al destino con un TTL de 1. El router del salto siguiente, 68.121.105.174, disminuye el TTL del paquete, encuentra que es cero, y dispara de regreso un mensaje de “icmp: time exceeded in-transit” (icmp: tiempo sobrepasado en tránsito). Esto lo hace tres veces el anfitrión A, de modo que se pueden ver tres impresiones diferentes de tiempo (7.689 ms, 8.471 ms y 8.556 ms). Con estas tres respuestas de regreso, el anfitrión A intenta de nuevo con otro paquete UDP hacia el destino, en esta ocasión con un TTL de 2 (usando una opción **-v** en la línea de **tcpdump** hubiéramos mostrado esto). El salto 68.121.105.174 hace el TTL 1, lo cual quiere decir que el segundo salto, 64.164.97.67, envía de regreso el mensaje ICMP. Esto se repite hasta que vemos un mensaje ICMP de “unreachable” (inalcanzable) que nos dice que hemos encontrado nuestro punto final.

¿Por qué es lento el DNS?

Problemas raros o intermitentes son grandes candidatos en el uso de **tcpdump**. Usando un rastro de los propios paquetes, puede mirar la actividad durante cierto periodo e identificar los aspectos que pueden estar enmascarados por otra actividad del sistema o por una falta de herramientas de eliminación de errores.

Supongamos por un momento que está usando el servidor DNS administrado por su proveedor DSL. Todo está funcionando, hasta que un día parece que algo anda mal. Específicamente, cuando visita un sitio Web, la primera conexión parece que tarda mucho tiempo, pero una vez conectado, el sistema parece funcionar con bastante rapidez. Cada par de sitios, la conexión incluso no trabaja, pero haciendo clic en “Reload” (recargar) parece que logra hacer el truco. Eso significa que el DNS está funcionando y la conectividad está allí. ¿Qué lo causa?

Es el momento para seguirle el rastro a un paquete. Como éste es un tráfico Web, sabemos que están trabajando dos protocolos: DNS para la resolución del nombre del anfitrión y TCP para el establecimiento de la conexión. Eso quiere decir que queremos filtrar todos los demás ruidos y enfocarnos en esos dos protocolos. Como parece que hay algún tipo de aspecto relacionado con la velocidad, es necesario obtener las impresiones de los tiempos del paquete, de modo que no queremos usar la opción **-t**. El resultado es

```
[root@hostA:~]# tcpdump -n port 80 or port 53
```

Ahora visite el sitio Web deseado. Para este ejemplo, iremos a www.rondcore.com.

Miremos unos cuantos de los primeros paquetes UDP:

```
21:27:40 68.12.10.17.4102 > 206.13.31.12.53: A? rondcore.com (31)
21:27:50 68.12.10.17.4103 > 206.13.31.12.53: A? rondcore.com (31)
21:27:58 206.13.31.12.53 > 68.12.10.17.4102: 1/4/4 A 67.43.6.47 (206)
```

Eso es interesante... necesitamos retransmitir la petición DNS con el fin de obtener la dirección IP para el nombre del anfitrión. Parece que aquí hay algún tipo de problema de conectividad, ya que llega el momento en que obtenemos la respuesta de regreso. ¿Qué hay acerca del resto de la conexión? ¿El problema de conectividad produce un impacto sobre otra actividad?

```
21:27:58 68.12.10.17.3013 > 67.43.6.47.80: S 1031:1031(0) win 57344 (DF)
21:27:58 67.43.6.47.80 > 68.12.10.17.3013: S 192:192(0) ack 1031 win 5840 (DF)
21:27:58 68.12.10.17.3013 > 67.43.6.47.80: . ack 1 win 58400 (DF)
21:27:58 68.12.10.17.3013 > 67.43.6.47.80: P 1:17(16) ack 1 win 58400 (DF)
21:27:58 67.43.6.47.80 > 68.12.10.17.3013: . ack 17 win 5840 (DF)
21:27:58 68.12.10.17.3013 > 67.43.6.47.80: P 17:94(77) ack 1 win 58400 (DF)
21:27:58 67.43.6.47.80 > 68.12.10.17.3013: . ack 94 win 5840 (DF)
21:27:58 67.43.6.47.80 > 68.12.10.17.3013: . 1:1461(1460) ack 94 win 5840 (DF)
21:27:58 67.43.6.47.80 > 68.12.10.17.3013: F 2155:2155(0) ack 94 win 5840 (DF)
21:27:58 68.12.10.17.3013 > 67.43.6.47.80: . ack 1461 win 56940 (DF)
21:27:58 67.43.6.47.80 > 68.12.10.17.3013: P 1461:2155(694) ack 94 win 5840 (DF)
21:27:58 68.12.10.17.3013 > 67.43.6.47.80: . ack 2156 win 56511 (DF)
21:27:58 68.12.10.17.3013 > 67.43.6.47.80: F 94:94(0) ack 2156 win 58400 (DF)
21:27:58 67.43.6.47.80 > 68.12.10.17.3013: . ack 95 win 5840 (DF)
```

Es evidente que el resto de la conexión se hizo con rapidez. El tiempo por empujar en el servidor DNS...

```
[sshah@hostA:~]$ ping 206.13.28.12
PING 206.13.28.12 (206.13.28.12) from 192.168.1.15 : 56(84) bytes of data.
64 bytes from 206.13.28.12: icmp_seq=1 ttl=247 time=213.0 ms
64 bytes from 206.13.28.12: icmp_seq=3 ttl=247 time=477.0 ms
64 bytes from 206.13.28.12: icmp_seq=4 ttl=247 time=177.5 ms
```

¡Claro! Estamos perdiendo paquetes y la intranquilidad en el alambre es muy mala. Esto explica el singular comportamiento de DNS. Es momento de buscar otro servidor DNS mientras este asunto se resuelve.

Trazo de gráficas de singularidades y fines

Como administradores, a veces tenemos preguntas acerca del sistema que son, bien, sólo por el deseo de preguntar. Cuando llega el momento de reunir información sobre la red, **tcpdump** es una mina de oro. Enseguida tiene unos cuantos ejemplos de las cosas que puede hacer.

Representación gráfica de los números iniciales de secuencia

El Initial Sequence Number (ISN, Número inicial de secuencia) en una conexión TCP es el número de secuencia especificado en el paquete SYN que inicia una conexión. Por razones de seguridad, es importante contar con un ISN suficientemente aleatorio de modo que otros no puedan embromar las conexiones con su servidor. Para ver una gráfica de la distribución de los ISN que su servidor está generando, usemos **tcpdump** para capturar los paquetes SYN/ACK enviados desde el servidor Web. Para capturar los datos, usemos el bit siguiente de **tcpdump** enviado por tubería a Perl:

```
[root@hostA:~]# tcpdump -l -n -t "tcp[13] == 18" | perl -ane  
'($s,$j)=split(/:/,$F[4],2); print "$s\n";' > graphme
```

El comando **tcpdump** introduce un nuevo parámetro, **-l**. Este parámetro le dice a **tcpdump** que ponga en línea memoria intermedia en su salida. Esto es necesario cuando se envía por tubería la salida de **tcpdump** hacia otro programa, como Perl. También introdujimos un nuevo truco, mediante el cual miramos una desviación específica en bytes del paquete TCP y verificamos su valor. En este caso, usamos la cifra del encabezado TCP con el fin de determinar que el byte 13º cumple con las banderas TCP. Para SYN/ACK, el valor es 18. La línea resultante se envía por tubería a un script Perl que extrae el número de secuencia de la línea y lo imprime. El archivo resultante **graphme** sencillamente será una cadena de números que se mira de manera semejante a esto:

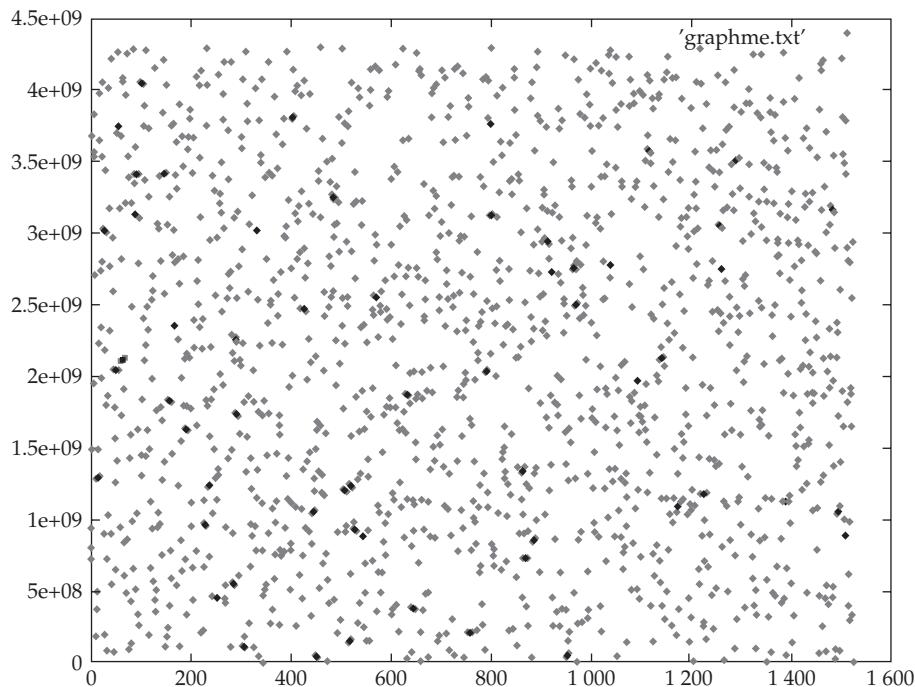
```
803950992  
1953034072  
3833050563  
3564335347  
2706314477
```

Ahora usamos **gnuplot** (<http://www.gnuplot.info>) para construir la gráfica de éstos. El lector podría usar otra hoja de cálculo para construir la gráfica, pero dependiendo de cuántas entradas tiene, eso podría ser algo a considerar. El programa **gnuplot** funciona bien con conjuntos grandes de datos y es gratis.

Iniciamos **gnuplot** y emitimos los comandos siguientes:

```
[sshah@hostA:~]$ gnuplot  
gnuplot> set terminal png  
Terminal type set to 'png'  
Options are 'small monochrome'  
gnuplot> set output 'syns.png'  
gnuplot> plot 'graphme'  
gnuplot> quit
```

Echando una mirada al archivo `syns.png`, vemos la gráfica siguiente:



La gráfica muestra una buena distribución de valores ISN. Esto implica que es difícil embromar las conexiones TCP para este anfitrión. Es evidente que entre más datos tenga la gráfica que se da aquí, más seguro puede estar usted de este resultado. Llevar los datos hacia un paquete de estadística con el fin de confirmar el resultado puede ser igualmente interesante.

¿Qué es mi tamaño promedio de paquete?

La magnitud del rendimiento que logra una red tiene mucho que ver con los tamaños de los paquetes. Entre más contenga un paquete, más eficiente es, debido a los costos fijos de tratar con uno de ellos. Los académicos han encontrado que la distribución de los tamaños de los paquetes es típicamente bimodal, con una punta en torno a los paquetes pequeños de 64 bytes (por lo común, debido a los reconocimientos TCP) y a los paquetes grandes (paquetes TCP que empujan datos reales).

Los académicos pueden haber hecho su propia investigación, pero siempre es una buena diversión averiguar si se aplica a su servidor. Para obtener la misma información, use `tcpdump` y un poquito de `awk`. Para el no iniciado, `awk` es otro lenguaje para scripts que viene como algo estándar con la mayor parte de los sistemas UNIX y Linux. Para las máquinas que no son Linux, a menudo Perl es opcional; sin embargo, está garantizado que `awk` está allí. Para obtener más información acerca de `awk`, visite <http://www.gnu.org/software/gawk/gawk.html>.

Empecemos con el comando **tcpdump** para reunir los datos. Como en la sección anterior, usamos el parámetro **-l** para hacer que la línea de salida se almacene en memoria intermedia y entubamos la salida. Sin embargo, en esta ocasión la elección de **awk** está lejos de ser ociosa; se requieren menos caracteres expresar lo que queremos con **awk** que con Perl.

```
[root@hostA:~]# tcpdump -l -q -t -n | awk '{print $5}' > psize
```

Dependiendo de lo que está buscando, puede ser que quiera alterar el filtro en **tcpdump** con el fin de aislar los tipos separados de tráfico. Por ejemplo, si sólo está interesado en el tráfico Web, la aplicación de un filtro “port 80” hará lo necesario. Esto también es un ejemplo excelente de en dónde, por el contrario, el uso del parámetro **-w** para escribir todo el tráfico sería una buena selección. Con los paquetes en bruto almacenados en el disco, puede regresar y aplicar diferentes filtros, múltiples veces, a los mismos datos para obtener distintos tipos de gráficas. Por ejemplo, ¿existe alguna diferencia en la distribución entre el tráfico Web y el de correo? ¿Qué se puede decir acerca del tráfico que no es TCP en comparación con el que sí lo es? Por el momento, supongamos que queremos ver todo el tráfico y miramos el archivo **packet_size.txt**.

Lo que veremos es una lista de números entre 0 y 1 460, que muestra cuántos datos están en cada paquete. ¿Por qué 1 460? Porque la salida no está tomando en cuenta los encabezados de los propios paquetes, y los encabezados TCP + IP son 40 bytes como mínimo. Con la lista a la mano, necesitamos ordenarlos y contarlos. Para hacer esto, usamos los comandos **sort** y **uniq**, como sigue:

```
[sshah@hostA:~]$ sort -n psize | uniq -c | awk '{print $2,$1}' > psize2
```

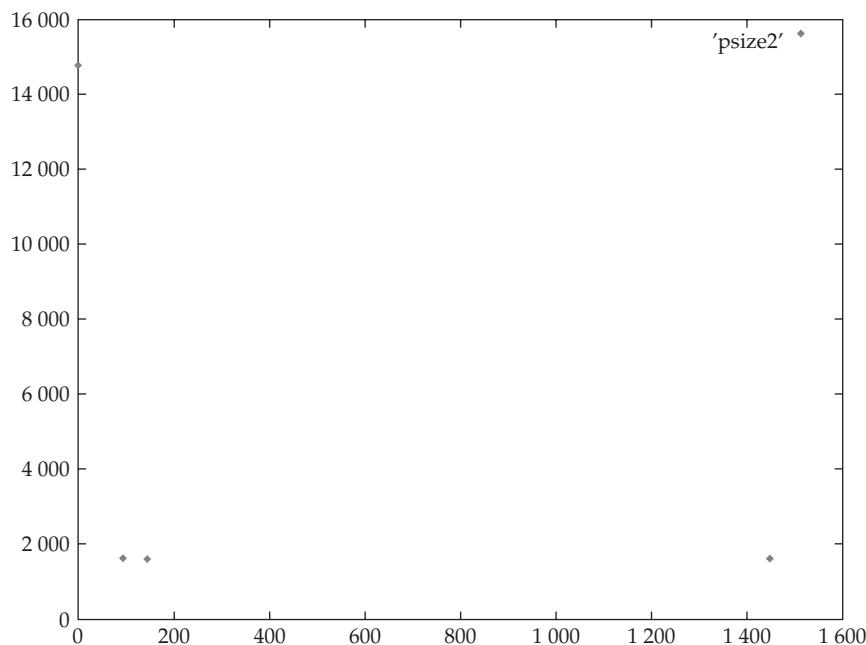
El comando **sort** hace una clasificación numérica de todos los números, de modo que los 0 quedan agrupados, los 1 también, y así sucesivamente. Esto se entuba hacia **uniq**, el cual cuenta cuántos hay de cada número y presenta el conteo seguido por el número. Para obtener nuestro eje correcto, esta salida se entuba hacia **awk**, de modo que se inviertan los números. El valor del eje x (el primer número) es el tamaño del paquete, seguido por cuantos hay.

Un rápido recorrido por **gnuplot** sería

```
[sshah@hostA:~]$ gnuplot
gnuplot> set terminal png
Terminal type set to 'png'
Options are 'small monochrome'
gnuplot> set output 'packet_size.png'
gnuplot> plot 'psize2'
gnuplot> quit
```

Para mi servidor de prueba particular, ejecutando una herramienta de generación de carga (Apache Benchmark, vea la página man para “ab”) contra una pequeña página inicial produjo una gran cantidad de paquetes pequeños. Es evidente que la mayor parte de los paquetes tenían cero bytes de longitud, lo cual significaba que estuvo presente una gran cantidad de carga general

en relación con el montaje y el desmontaje. El resultado se muestra enseguida (la gráfica de usted debe ser un poco diferente):



RESUMEN

En este capítulo se cubrieron los fundamentos de TCP/IP y otros protocolos, ARP, la creación de subredes y las netmasks, así como el encaminamiento. Es mucho para digerir. Pero tenemos la esperanza de que esta versión simplificada deba facilitar su comprensión. En específico, discutimos:

- ▼ Cómo se relaciona TCP/IP con el modelo de siete capas ISO OSI
- La composición de un paquete
- Las especificaciones de los encabezados de los paquetes y la manera de obtenerlos usando la herramienta **tcpdump**
- El proceso completo de establecimiento de una conexión TCP, la transferencia de datos y el desmontaje de la conexión
- Cómo calcular las netmasks
- Cómo funciona el encaminamiento estático
- Cómo funciona el encaminamiento dinámico con RIP
- ▲ Varios ejemplos del uso de **tcpdump**

Debido a que la información que se da en este texto está (sustancialmente) simplificada respecto de la real, puede ser que el lector quiera echar una ojeada a algunos otros libros para obtener más información referente a este tema. Esto resulta en especial de importancia si tiene redes complejas en las que sus máquinas necesitan vivir o si necesita entender la operación del firewall.

Un libro que recomendamos a todos es *TCP/IP Illustrated, Volume 1*, por Richard Stevens (Addison-Wesley, 1994). En este libro se cubre TCP/IP con profundidad, así como varios protocolos que envían sus datos por IP. Stevens realiza un trabajo fantástico de explicación de este complejo tema de una manera muy clara y metódica. Todavía no hemos encontrado una sola persona que lo haya leído y no entienda lo que el autor dijo. Un administrador de sistema que haya leído este libro se puede hacer con facilidad mucho más valioso en el mercado de trabajo.

Si necesita empezar con algo un poco menos sustancial, pruebe con *TCP/IP for Dummies, Fifth Edition*, por Candace Leiden y otros (Hungry Minds, 2003). Aun cuando es un libro para "ignorantes", le agradará ver que la cobertura en realidad es profunda, sólo que con más lentitud. Y, afortunadamente, los autores no resbalan hacia lo correcto para hacerlo más sencillo. (¡A los muchachos de mercadeo y ventas que necesitan vender hardware para operación en red se les debe exigir que lean este libro!)

Mi última recomendación es específica para quienes necesitan configurar equipo Cisco (lo cual según las cifras de ventas de Cisco, ¡son una gran cantidad de ustedes!): el *Cisco TCP/IP Routing Professional Reference*, por Chris Lewis (McGraw-Hill/Osborne, 2000). Chris realiza un trabajo sólido en la explicación tanto de los protocolos como de los aspectos de configuración para el equipo Cisco.

CAPÍTULO 12



Configuración
de la red

Saber cómo configurar sus servicios de red a mano puede tener una terrible importancia por varias razones. La primera y principal es que cuando las cosas se están descomponiendo y no puedes iniciar su GUI favorita, resulta crucial ser capaz de manejar la configuración de la red desde la línea de comandos. La otra razón es la administración remota: puede ser que no sea capaz de ejecutar una herramienta gráfica de configuración desde un sitio remoto. Aspectos como los firewalls y la latencia de la red es probable que restrinjan su administración remota sólo a la línea de comandos. Por último, siempre es bueno poder realizar la configuración de la red a través de scripts, y las herramientas de la línea de comandos son las más apropiadas para el uso de éstos.

En este capítulo obtendremos una perspectiva general de los controladores de interfaz de redes; es decir, las dos herramientas necesarias para realizar la administración por medio de la línea de comandos de su interfaz de redes: **ifconfig** y **route**.

MÓDULOS E INTERFACES DE REDES

Bajo Linux, los dispositivos de redes rompen la tradición de tener acceso a los dispositivos a través de la metáfora archivo. No es hasta que el controlador de redes inicializa la tarjeta y los propios registradores con el núcleo que en realidad existe un mecanismo para que cualquiera tenga acceso a la tarjeta. Por lo general, los dispositivos de Ethernet se registran por sí mismos como ethX, en donde X es el número de dispositivo. El primer dispositivo Ethernet es el eth0, el segundo es eth1, etcétera.

Dependiendo de la manera en que se compiló el núcleo de usted, los controladores de dispositivos para sus tarjetas de interfaz de redes pueden haber sido compilados como un módulo. Para la mayor parte de las distribuciones, éste es el mecanismo predeterminado para el embarque, ya que hace mucho más fácil sondear el hardware.

Si el controlador se configura como un módulo y usted ha establecido la autocarga de los módulos, necesitará decirle al núcleo la conversión entre los nombres de dispositivos y el módulo, con el fin de cargar en el archivo **/etc/modprobe.conf**. Por ejemplo, si su dispositivo eth0 es una tarjeta Intel PRO/1000, agregaría la línea siguiente a su archivo **/etc/modprobe.conf**:

```
alias eth0 e1000
```

en donde **e1000** es el nombre del controlador de dispositivo.

Necesitará fijar esto para todas las tarjetas de redes que tenga en el mismo sistema. Por ejemplo, si tiene dos tarjetas de redes, una basada en el conjunto de chips DEC Tulip y la otra en el RealTek 8169, necesitaría asegurarse de que su archivo **/etc/modprobe.conf** incluye estas líneas:

```
alias eth0 tulip
alias eth1 r8169
```

en donde **tulip** se refiere a la tarjeta de redes con el chip Tulip en ella y **r8169** se refiere a la RealTek 8169.

NOTA Estos comandos de alias no serán las únicas entradas en el **/etc/modprobe.conf**.

En el directorio `/lib/modules/`uname -r`/kernel/drivers/net` puede encontrar una lista de todos los controladores de dispositivos que están instalados para su núcleo, de este modo:

```
[root@hostA etc]# cd /lib/modules/`uname -r`/kernel/drivers/net  
[root@hostA net]#
```

Note que existen acentos inversos rodeando al comando `uname -r` incrustado. Esto le permitirá asegurarse de que está usando la versión correcta del controlador para su versión actual del núcleo. Si está usando una instalación estándar de su distribución, encontrará que sólo debe haber un nombre de subdirectorio en el directorio `/lib/modules`.

Si quiere ver la descripción de un controlador, sin tener que cargar el propio controlador, use el comando `strings` del archivo `.ko`. Por ejemplo, para ver la descripción del controlador `yellowfin.ko`, teclee

```
[root@hostA net]# strings yellowfin.ko | grep description=  
description=Packet Engines Yellowfin G-NIC Gigabit Ethernet driver
```

Para obtener una lista de todas las descripciones de controladores, puede usar el siguiente script de shell:

```
#!/bin/sh  
KERNEL=`uname -r`  
for i in `find /lib/modules/$KERNEL/kernel/drivers/net -name "*.ko" -print`  
do  
    DESCRIPTION=`strings $i | grep 'description=' | sed 's/description=/ /'  
    echo $i "—" $DESCRIPTION  
done
```

Tenga presente que no todos los controladores tienen descripciones asociadas con ellos, pero la mayor parte sí la tienen. Cuando copie este script, tenga cuidado de advertir las comillas sencillas, las comillas dobles, los espacios y los acentos inversos.

USO DE IFCONFIG PARA CONFIGURAR DIRECCIONES IP ASIGNADAS

El programa `ifconfig` es el responsable de fijar los valores de sus tarjetas de interfaz de redes (NIC, *network interface cards*). Todas sus operaciones se pueden realizar a través de opciones de la línea de comandos, ya que su formato nativo no tiene menús ni interfaz gráfica. Los administradores que han usado el programa `ifconfig` de Windows pueden ver algunas semejanzas, ya que Microsoft implementó algunas herramientas CLI para operación en red que remedaron los subconjuntos funcionales de sus hermanos UNIX.

SUGERENCIA Los administradores que todavía tratan con Windows pueden hallar que el programa `%SYSTEMROOT%\system32\netsh.exe` resulta una herramienta práctica para exponer los detalles de las redes Windows a través de la CLI, por medio de una interfaz semejante a Cisco.

NOTA Por lo general, el programa **ifconfig** reside en el directorio **/sbin** y se debe ejecutar como raíz. Algunos scripts de conexión, como los que se encuentran en Fedora, no incluyen de manera predeterminada **/sbin** en la PATH. Por consiguiente, puede ser que necesite solicitar **/sbin/ifconfig** al llamar ese directorio. Si espera ser un usuario frecuente de los comandos **/sbin**, puede ser que encuentre prudente agregar **/sbin** a su PATH.

Se han escrito varias herramientas para envolver la interfaz de línea de comandos de **ifconfig** para proporcionar interfaces manejadas por menús o gráficas, y muchas de estas herramientas se embarcan con las distribuciones de Linux. Por ejemplo, Fedora tiene una herramienta GUI bajo el menú Applications | System Settings (Aplicaciones | Ajustes del sistema).

Como administrador, debe por lo menos saber cómo configurar la interfaz de redes a mano; saber cómo hacerlo es inestimable, ya que muchas opciones adicionales que no se muestran en las GUI se exponen en la CLI. Por esa razón, en esta sección se cubrirá el uso de la herramienta de línea de comandos **ifconfig**.

Uso simple

En su uso más sencillo, todo lo que necesitará hacer es proporcionar el nombre de la interfaz que se está configurando y la dirección IP. El programa **ifconfig** deducirá el resto de la información a partir de esta última. Como consecuencia, podría introducir

```
[root@hostA /root]# ifconfig eth0 192.168.1.42
```

para fijar el dispositivo eth0 a la dirección IP 192.168.1.42. Debido a que la 192.168.1.42 es una dirección clase C, la netmask calculada será 255.255.255.0 y la dirección de difusión será 192.168.1.255.

Si la dirección IP que está fijando es una de clase A o B, que se envía a las subredes de modo diferente, necesitará fijar de modo explícito las direcciones de difusión y de netmask en la línea de comandos, como sigue:

```
[root@hostA /root]# ifconfig dev ip netmask nmask broadcast bcast
```

en donde **dev** es el dispositivo de red que está usted configurando, **ip** es la dirección IP a la que se lo está asignando, **nmask** es la netmask y **bcast** es la dirección de difusión. Por ejemplo, lo siguiente asignará el dispositivo eth0 a la dirección IP 1.1.1.1, con una netmask de 255.255.255.0 y una dirección de difusión de 1.1.1.255:

```
[root@hostA /root]# ifconfig eth0 1.1.1.1 netmask 255.255.255.0 broadcast  
1.1.1.255
```

SUGERENCIA Puede hacer una lista de los dispositivos activos al ejecutar **ifconfig** sin parámetros. Puede hacer una lista de todos los dispositivos, sin importar si están activos, al ejecutar **ifconfig -a**.

Asignación de alias IP

En algunos casos, es necesario que un solo anfitrión tenga múltiples direcciones IP. Linux puede soportar esto mediante el uso de alias IP.

Cada interfaz del sistema Linux puede tener asignadas múltiples direcciones IP. Esto se hace enumerando cada caso de la misma interfaz con dos puntos seguidos por un número. Por ejemplo, eth0 es la interfaz principal, eth0:0 es un alias de esa interfaz, eth0:1 es un alias de la misma interfaz, etcétera.

Configurar un alias de una interfaz es sólo como configurar cualquier otra interfaz: sencillamente use **ifconfig**. Por ejemplo, para asignar a eth0:0 la dirección 10.0.0.2 y netmask 255.255.255.0, haríamos lo siguiente:

```
[root@hostA /root]# ifconfig eth0:0 10.0.0.2 netmask 255.255.255.0
[root@hostA /root]# ifconfig eth0:0
eth0:0      Link encap:Ethernet HWaddr 00:30:48:21:2A:36
            inet addr:10.0.0.2 Bcast:10.255.255.255 Mask:255.255.0.0
                      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                      Interrupt:9 Base address:0xd000
```

Note que las conexiones de la red hicieron que el alias de la interfaz *se* comunicara usando su dirección IP alias; sin embargo, en la mayor parte de las circunstancias, cualquier conexión que se origina del anfitrión hacia otro anfitrión usará la primera IP asignada de la interfaz. Por ejemplo, si eth0 es 192.168.1.15 y eth0:0 es 10.0.0.2, una conexión de la máquina que se encamina por eth0 usará la dirección IP 192.168.1.15. La excepción para este comportamiento es para las aplicaciones que se ligan a una dirección IP específica. En esos casos es posible que la aplicación origine conexiones desde la dirección IP alias. En el caso de que un anfitrión tenga múltiples interfaces, la tabla de encaminamiento decidirá cuál interfaz usar. Con base en la información de encaminamiento, se usará la primera dirección IP asignada de la interfaz.

¿Confuso? No se preocupe, es un poco inusual captar la idea al principio. La elección de la IP fuente se asocia también al encaminamiento, de modo que volveremos a visitar este concepto más adelante en el capítulo.

Estructuración de las NIC en el momento de la inicialización

Por desgracia, cada distribución ha tomado de manera un poco diferente el proceso de automatizar su proceso de estructuración para las tarjetas de redes. En la sección que sigue cubriremos los aspectos específicos de Fedora Core 3. Para otras distribuciones necesita manejar este procedimiento en una de dos maneras:

- ▼ Usar la herramienta de administración que viene con esa distribución para agregar el soporte de tarjetas de redes. Probablemente éste es el proceso más fácil y más confiable.
- ▲ Hallar el script que es responsable de la configuración de las tarjetas de redes (funciona bien usar la herramienta **grep** para hallar cuál script ejecuta **ifconfig**). Al final del script agregue las instrucciones necesarias para **ifconfig**. Otro lugar para agregar las instrucciones necesarias es en el script **rc.local**; no es tan bonito, pero funciona igualmente bien.

Estructuración de las NIC bajo Fedora Core 3 y la Red Hat Enterprise

Fedora Core y Red Hat Enterprise tienen un sistema de montaje que facilita la configuración de las tarjetas de redes en el momento de la inicialización. Se hace a través de la creación de archivos en el directorio **/etc/sysconfig/network-scripts** que se leen en el momento del arranque. Todas

las herramientas gráficas bajo Linux crean estos archivos para usted, pero para el usuario poderoso que se encuentra en el interior de usted, muriéndose sólo por editar los archivos a mano, lo siguiente es lo que necesita saber.

Para cada interfaz de redes existe un archivo **ifcfg** en **/etc/sysconfig/network-scripts**. Este nombre de archivo trae el sufijo del nombre del dispositivo; por tanto, **ifcfg-eth0** es para el dispositivo **eth0**, **ifcfg-eth1** es para el **eth1**, y así sucesivamente.

Si elige un IP estático en el momento de la instalación, el formato de cada uno de estos archivos será como el siguiente:

```
DEVICE="eth0"
IPADDR="192.168.1.1"
NETMASK="255.255.255.0"
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
ONBOOT="yes"
BOOTPROTO="static"
```

SUGERENCIA A veces, si está ejecutando otros protocolos, por ejemplo IPX, podría ver variables que empiezan con IPX. Si no tiene que ejecutar IPX (lo cual es típico), puede eliminar con seguridad las líneas que tiene IPX en ellas.

Si elige usar DHCP en el momento de la instalación, su archivo se verá como el siguiente:

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
TYPE=Ethernet
```

Estos campos determinan la información de la configuración de IP para el dispositivo **eth0**. Advierta cómo cada uno de estos valores corresponde a los parámetros de **ifconfig**. Para cambiar la información de configuración para este dispositivo, sencillamente cambie la información en el archivo **ifcfg** y ejecute

```
[root@hostA /root]# cd /etc/sysconfig/network-scripts
[root@hostA network-scripts]# ./ifdown eth0
[root@hostA network-scripts]# ./ifup eth0
```

Si está cambiando de DHCP a una dirección IP estática sólo necesita cambiar **BOOTPROTO** que sea igual a "yes" y agregar las líneas para **IPADDR**, **NETWORK** y **BROADCAST**. Si necesita configurar una segunda tarjeta de interfaz de redes, copie el archivo **ifcfg-eth0** en el **ifcfg-eth1** y cambie la información en este último para que refleje la información de la segunda tarjeta. Una vez allí, Red Hat y Fedora Core la configurarán de manera automática en el curso de la inicialización siguiente.

Si necesita activar la tarjeta de inmediato, ejecute

```
[root@hostA /root]# cd /etc/sysconfig/network-scripts
[root@hostA network-scripts]# ./ifup eth1
```

NOTA Es posible configurar direcciones IP alias usando también este método.

Parámetros adicionales

El formato del comando **ifconfig** es como sigue:

```
[root@hostA /root]# ifconfig device address options
```

en donde **device** es el nombre del dispositivo Ethernet (por ejemplo, **eth0**), **address** es la dirección IP que desea aplicar al dispositivo y **options** es una de las siguientes:

Opción	Descripción
up	Activa el dispositivo. Esta opción es implícita.
down	Desactiva el dispositivo.
arp	Activa este dispositivo para dar respuesta a peticiones arp (predeterminado).
-arp	Desactiva este dispositivo para que no dé respuesta a peticiones arp .
mtu value	Fija la unidad máxima de transmisión (MTU) del dispositivo en un value . Bajo Ethernet, este valor predeterminado es 1 500. (Vea la nota que sigue a esta tabla, referente a ciertas tarjetas Gigabit Ethernet.)
netmask address	Fija la netmask para esta interfaz en address . Si no se proporciona un valor, ifconfig calcula esta netmask a partir de la clase de la dirección IP. A una clase A se le da una netmask de 255.0.0.0, a la B una de 255.255.0.0 y a la C una de 255.255.255.0.
broadcast address	Fija la dirección de difusión para esta interfaz en address . Si no se proporciona un valor, ifconfig calcula esta dirección a partir de la clase de la dirección IP, de manera semejante a la netmask.
pointtopoint address	Establece una conexión punto a punto (PPP) en donde la dirección remota es address .

NOTA Muchas tarjetas Gigabit Ethernet ahora soportan marcos Ethernet jumbo. Un marco jumbo tiene 9 000 bytes de longitud, lo cual (en forma conveniente) contiene un paquete NFS completo. Esto permite a los servidores de archivos comportarse mejor, ya que tienen que consumir menos tiempo fragmentando paquetes para acomodarlos en marcos Ethernet de 1 500 bytes. Por supuesto, la infraestructura de su red como un todo debe soportar esto para lograr el beneficio. Si tiene una tarjeta de redes y un hardware apropiado de red para estructurar marcos jumbo, resulta de mucha utilidad ver en qué forma se comután esas características. Si su tarjeta Gigabit Ethernet lo soporta, puede fijar el tamaño del marco hacia 9 000 bytes al cambiar el valor de la MTU cuando se realiza la configuración con **ifconfig** (por ejemplo, **ifconfig eth0 192.168.1.1 mtu=9000**).

Existen muchas más opciones para **ifconfig**; sin embargo, encontrará que es desacostumbrado necesitar más de ellas. Para los curiosos, sencillamente lean la página de manual para **ifconfig** con el fin de ver la lista completa de las opciones con sus descripciones.

USO DE LAS RUTAS

Si su anfitrión está conectado a una red con múltiples subredes, necesita un *router* o *puerta de acceso*. Este dispositivo, el cual se asienta entre las redes, redirige los paquetes hacia su destino real (por lo común, la mayor parte de los anfitriones no conoce la trayectoria correcta hacia un destino; sólo conocen el propio destino).

En el caso en el que un anfitrión incluso no cuenta con el primer indicio acerca de hacia dónde enviar el paquete, usa su *ruta predeterminada*. Esta trayectoria apunta hacia un router, el cual, de manera ideal, sí tiene idea de hacia dónde debe ir un paquete o, por lo menos, sabe de otro router que puede tomar decisiones más inteligentes.

NOTA En los sistemas Red Hat y Fedora Core, la ruta predeterminada suele estar almacenada como la variable **GATEWAY** en el archivo apropiado de la interfaz, en **/etc/sysconfig/network**.

Un anfitrión Linux típico conoce tres rutas: la primera es el bucle cerrado, el cual sencillamente apunta hacia el dispositivo de bucle cerrado. La segunda es la ruta de la red de área local, de modo que los paquetes destinados a anfitriones que están dentro de la misma LAN se envían en forma directa a ellos. Por último, la tercera es la predeterminada. Esta ruta se usa para paquetes que necesitan salir de la red de área local con el fin de comunicarse con otras redes.

Si estructura la configuración de su red en el momento de la instalación, lo más probable es que este ajuste ya se tome en cuenta por usted, de modo que no necesita cambiarlo. Sin embargo, esto no quiere decir que no pueda.

NOTA En realidad, hay casos en los cuales necesitará cambiar sus rutas a mano. Por lo general, esto es necesario cuando se instalan múltiples tarjetas en el mismo anfitrión, en donde cada NIC se conecta a una red diferente. El lector debe saber cómo agregar una ruta, de modo que se puedan enviar los paquetes hacia la red apropiada para una dirección destino dada.

Uso simple

El comando típico **route** se estructura como sigue:

```
[root@hostA /root]# route cmd type addy netmask mask gw gway dev dn
```

Los parámetros son los siguientes:

Parámetro	Descripción
cmd	add o del , dependiendo de si está agregando o borrando una ruta. Si está borrando una ruta, el único otro parámetro que necesita es addy .
type	-net o -host , dependiendo de si addy representa una dirección de la red o una de un router.
addy	La red destino a la cual quiere ofrecerle una ruta.
netmask mask	Fija la netmask a la dirección addy para la mask .
gw gway	Fija la dirección del router para addy en gway . Por lo común, usado para la ruta predeterminada.
dev dn	Envía todos los paquetes destinados a addy a través del dispositivo de la red dn según lo fija ifconfig .

Enseguida se muestra la manera de fijar la ruta predeterminada en un anfitrión muestra, el cual tiene un solo dispositivo Ethernet y un router de 192.168.1.1:

```
[root@hostA /root]# route add -net default gw 192.168.1.1 dev eth0
```

Esta línea de comandos estructura un sistema de modo que todos los paquetes destinados a 192.168.1.42 se envíen a través del primer dispositivo PPP:

```
[root@hostA /root]# route add -host 192.168.1.42 netmask 255.255.255.255 dev ppp0
```

Ahora, se muestra cómo borrar la ruta destinada a 192.168.1.42:

```
[root@hostA /root]# route del 192.168.1.42
```

NOTA Si está usando una puerta de acceso, necesita asegurarse de que existe una ruta hacia esa puerta, antes de hacer referencia a otra ruta. Por ejemplo, si su ruta predeterminada usa la puerta de acceso en 192.168.1.1, necesita asegurarse de que tiene una ruta para llegar primero a la red 192.168.1.0.

Presentación de las rutas

Existen dos maneras en que puede presentar su tabla de rutas: el comando **route** y **netstat**.

Route

El uso de **route** es la manera más fácil de presentar su tabla de rutas; sencillamente, ejecute **route** sin parámetros. Aquí tiene una ejecución completa, junto con la salida:

```
[root@hostA /root]# route
Kernel IP routing table
Destination     Gateway      Genmask       Flags Metric Ref Use Iface
10.10.2.0        *          255.255.255.0 UH    0      0   0   eth1
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         firewall    0.0.0.0     UG    0      0   0   eth1
```

Ve dos redes. La primera es la 192.168.1.0, la cual es accesible a través del primer dispositivo Ethernet, eth0. La segunda es la 10.10.2.0, la cual se conecta a través del segundo dispositivo Ethernet, eth1. La ruta predeterminada es 10.10.2.4; sin embargo, debido a que, en DNS, la dirección IP se resuelve hacia el firewall del nombre del anfitrión, **route** imprime su nombre del anfitrión, en lugar de la dirección IP.

Ya hemos discutido el destino, la puerta de acceso, la netmask (mencionada como **genmask** en esta tabla) y la **iface** (interfaz, fijada por la opción **dev** en **route**). Las otras entradas en la tabla tienen los siguientes significados:

Entrada	Descripción
flags	Un resumen del estado de la conexión, en donde cada letra tiene un significado: U La conexión es up. H El destino es un anfitrión. G El destino es una puerta de acceso.
metric	El costo de una ruta, por lo general medido en saltos. Esto tiene significado para los sistemas que tienen múltiples rutas para llegar al mismo destino, pero se prefiere una trayectoria sobre las otras. Suele preferirse una trayectoria con la métrica menor. En el núcleo Linux no se usa esta información, pero en ciertos protocolos avanzados de encaminamiento sí se hace.
ref	El número de referencias para esta ruta. Esto no se usa en el núcleo de Linux. Se encuentra aquí porque la propia herramienta de encaminamiento es una plataforma cruzada. Por tanto, se imprime este valor, ya que otros sistemas operativos sí lo usan.
use	El número de bucles cerrados caché con éxito en la ruta. Para ver este valor, use la opción -F al llamar route .

Advierta que **route** presentó los nombres de los anfitriones para cualesquiera direcciones IP que pudo ver y resolver. Aun cuando es bueno leer esto, se presenta un problema cuando se tienen dificultades en la red, y quedan no disponibles los servidores DNS o NIS. El comando **route** se colgará, tratando de resolver los nombres de los anfitriones y esperando ver si regresan los servidores y los resuelven. Esta espera se tomará varios minutos hasta que se consuma el tiempo de la petición.

Para darle vuelta a esto, use la opción **-n** con **route**, de modo que se muestre la misma información, pero **route** intentará realizar la resolución del nombre del anfitrión en las direcciones IP.

netstat

Normalmente el programa **netstat** se usa para presentar el estado de todas las conexiones de la red en un anfitrión. Sin embargo, con la opción **-r**, también puede presentar la tabla de encañamiento del núcleo. El lector debe notar que la mayor parte de los otros sistemas operativos Linux requieren que use este método de visión de las rutas.

Enseguida, se da un ejemplo de llamado de **netstat -r** y su salida correspondiente:

```
[root@hostA /root]# netstat -r
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
default         192.168.1.1   0.0.0.0      UG     0      0      0      eth0
```

En este ejemplo ve una configuración sencilla. El anfitrión tiene una sola tarjeta de interfaz de redes, está conectado a la red 192.168.1.0 y tiene una puerta de acceso predeterminada fijada en 192.168.1.1.

Como el comando **route**, **netstat** también puede tomar el parámetro **-n**, de modo que no realice resolución del nombre del anfitrión.

ROUTER LINUX SENCILLO

Linux tiene un número impresionante de características de operación en red, incluyendo la capacidad de actuar como un router con todas las características. Para las redes pequeñas, que necesitan una red de bajo costo en donde no se requiere un rendimiento extremadamente elevado, una PC de costo bajo con unas cuantas tarjetas de redes puede funcionar bastante bien.

Siendo realista, un router Linux puede mover unos cuantos cientos de megabytes/s, dependiendo de la velocidad de la PC, el caché de la CPU, el tipo de NIC, las interfaces PCI y la velocidad del bus de entrada. Aun cuando esto no reemplaza un router de clase Cisco completo, puede satisfacer las necesidades de entornos más pequeños, en donde no se mueva una gran cantidad de tráfico a través del router y el costo es una consideración significativa (se supo que los sistemas UNIX completos actuaron como routers completos en los primeros centros de datos de Internet, antes que se moviera hacia el hardware el procesamiento de tres capas).

Encaminamiento con rutas estáticas

Supongamos que queremos configurar como router un sistema Linux con página inicial dual, como se muestra en la figura 12-1.

En esta red queremos encaminar los paquetes entre la red 192.168.1.0/24 y la red 192.168.2.0/24. La ruta predeterminada es a través del router 192.168.1.8, el cual está efectuando NAT para la Internet (en el capítulo 13 discutimos NAT con más detalles). Para todas las máquinas en la red

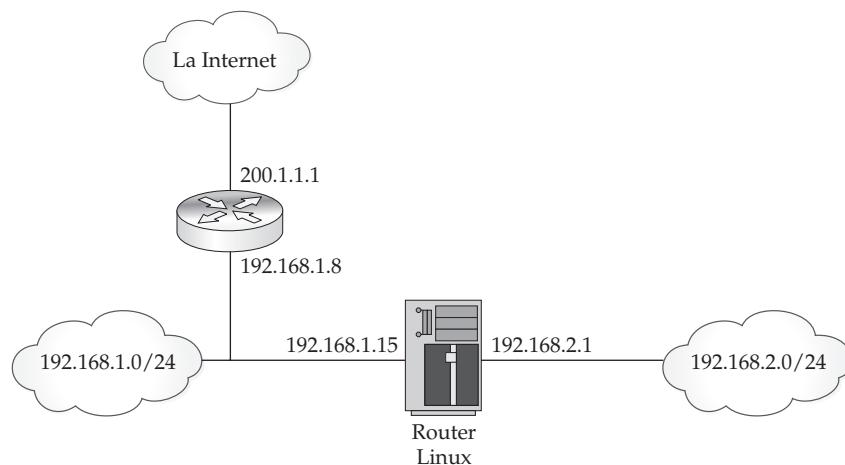


Figura 12-1. Nuestra red muestra

192.168.2.0/24, queremos solamente fijar su ruta predeterminada hacia 192.168.2.1, y dejar que el router Linux vea la manera de reenviar hacia la Internet y la red 192.168.1.0/24. Para los sistemas en esta última red, queremos configurar 192.168.1.15 como la ruta predeterminada, de modo que todas las máquinas puedan ver la Internet y la red 192.168.2.0/24.

Esto requiere que nuestro sistema Linux tenga dos interfaces de redes: eth0 y eth1. Configurémoslas como sigue:

```
[root@hostA /root]# ifconfig eth0 192.168.1.15 netmask 255.255.255.0
[root@hostA /root]# ifconfig eth1 192.168.2.1 netmask 255.255.255.0
```

El resultado se mira como esto:

```
[root@hostA /root]# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:30:48:21:2A:36
          inet addr:192.168.1.15  Bcast:192.168.1.255  Mask:255.255.255.0
                  UP BROADCAST RUNNING 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:1000
                  RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
                  Interrupt:9  Base address:0xd000

eth1      Link encap:Ethernet  HWaddr 00:02:B3:AC:5E:AC
          inet addr:192.168.2.1  Bcast:192.168.2.255  Mask:255.255.255.0
```

```

UP 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:1000
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
Base address:0xef80 Memory:febe0000-fec00000

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
              UP LOOPBACK RUNNING MTU:16436 Metric:1
              RX packets:2173386 errors:0 dropped:0 overruns:0 frame:0
              TX packets:2173386 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:0
              RX bytes:164613316 (156.9 Mb) TX bytes:164613316 (156.9 Mb)

```

NOTA Es posible configurar un router con un solo ramal, en donde la interfaz eth0 se configura con 192.168.1.15 y eth0:0 se configura con 192.168.2.1. Sin embargo, al hacer esto, se eliminará cualesquiera beneficios de segmentación de la red. En otras palabras, cualesquiera paquetes de difusión en el alambre serán vistos por las dos redes. Por consiguiente, suele preferirse poner cada red en su propia interfaz física.

Cuando **ifconfig** agrega una interfaz, también crea una entrada de ruta para esa interfaz, basada en el valor de la netmask. Así entonces, en el caso de la 192.168.1.0/24, se agrega una ruta en eth0 que envía todo el tráfico de esta red hacia ella. Con las dos interfaces de redes presentes, lancemos una mirada a la tabla de encaminamiento:

```
[root@hostA /root]# route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
192.168.2.0     0.0.0.0        255.255.255.0  U     0      0        0 eth1
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
```

Todo lo que está faltando aquí es la ruta predeterminada hacia 192.168.1.8. Agreguemos ésa con el uso del comando **route**.

```
[root@hostA /root]# route add default gw 192.168.1.8
[root@hostA /root]# route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
192.168.2.0     0.0.0.0        255.255.255.0  U     0      0        0 eth1
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          192.168.1.8   0.0.0.0      UG    0      0        0 eth0
```

Una verificación rápida con **ping** hace ver que tenemos conectividad a través de cada ruta:

```
[root@hostA /root]# ping -c 1 4.2.2.1
PING 4.2.2.1 (4.2.2.1) from 192.168.1.15 : 56(84) bytes of data.
64 bytes from 4.2.2.1: icmp_seq=1 ttl=245 time=15.2 ms

--- 4.2.2.1 ping statistics ---
1 packets transmitted, 1 received, 0% loss, time 0ms
rtt min/avg/max/mdev = 15.277/15.277/15.277/0.000 ms

[root@hostA /root]# ping -c 1 192.168.1.30
PING 192.168.1.30 (192.168.1.30) from 192.168.1.15 : 56(84) bytes of data.
64 bytes from 192.168.1.30: icmp_seq=1 ttl=64 time=0.233 ms

--- 192.168.1.30 ping statistics ---
1 packets transmitted, 1 received, 0% loss, time 0ms
rtt min/avg/max/mdev = 0.233/0.233/0.233/0.000 ms

[root@hostA /root]# ping -c 1 192.168.2.2
PING 192.168.2.2 (192.168.2.2) from 192.168.2.1 : 56(84) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=64 time=0.192 ms

--- 192.168.2.2 ping statistics ---
1 packets transmitted, 1 received, 0% loss, time 0ms
rtt min/avg/max/mdev = 0.192/0.192/0.192/0.000 ms
```

Se ve bien. Ahora es el momento de activar el reenvío IP. Esto le dice al núcleo de Linux que se le permite reenviar paquetes que no estén destinados a él, si tiene una ruta hacia el destino. Esto se hace al hacer **/proc/sys/net/ipv4/ip_forward** igual a 1, como sigue:

```
[root@hostA /root]# echo "1" > /proc/sys/net/ipv4/ip_forward
```

Debe fijarse la ruta predeterminada de los anfitriones en la red 192.168.1.0/24 hacia 192.168.1.15, y la de los anfitriones en la 192.168.2.0/24 debe fijarse hacia 192.168.2.1. Lo más importante, no olvide hacer que las adiciones de las rutas y la activación sean parte de **ip_forward** del script de arranque.

SUGERENCIA ¿Necesita un servidor DNS que no esté arriba de su cabeza? Para lograr una consulta rápida contra un servidor DNS externo, pruebe con 4.2.2.1, el cual en la actualidad le pertenece a Verizon. La dirección ha estado por allí durante mucho tiempo (originalmente perteneció a GTE Internet) y tiene números que son fáciles de recordar. Sin embargo, sea bueno acerca de ello: una o dos consultas rápidas para probar la conectividad está bien, pero hacerlo su servidor DNS primario no lo está.

CÓMO LINUX ELIGE UNA DIRECCIÓN IP

Ahora que el anfitrión A tiene dos interfaces (192.168.1.15 y 192.168.2.1), además de la interfaz de bucle cerrado (127.0.0.1), podemos observar cómo elegirá Linux una dirección IP fuente para comunicarse con ella.

Cuando se inicia una aplicación, tiene la opción de ligarse con una dirección IP. Si la aplicación no lo hace así explícitamente, Linux de manera automática elegirá la dirección IP en beneficio de esa aplicación en términos de conexión por conexión. Cuando Linux está tomando la decisión, examina la dirección IP destino de una conexión, toma la decisión de un encaminamiento con base en la tabla de encaminamiento actual y, a continuación, selecciona la dirección IP correspondiente a la interfaz de la que saldrá la conexión. Por ejemplo, si una aplicación en el anfitrión A hace una conexión hacia 192.168.1.100, Linux encontrará que el paquete debe salir de la interfaz eth0 y, por tanto, la dirección IP fuente para la conexión será 192.168.1.15.

Supongamos que la aplicación sí elige ligarse con una dirección IP. Si la aplicación fuera a ligarse con 192.168.2.1, Linux elegirá la fuente como la de la dirección IP fuente, sin importar de cuál interfaz saldrá la conexión. Por ejemplo, si la aplicación se liga a 192.168.2.1 y se hace una conexión hacia 192.168.1.100, la conexión saldrá de eth0 (192.168.1.15) con la dirección IP fuente de 192.168.2.1. Ahora es responsabilidad del anfitrión remoto (192.168.1.100) saber cómo enviar un paquete de regreso a 192.168.2.1 (es de presumir que la ruta predeterminada para 192.168.1.100 le hará saber a éste cómo tratar con ese caso).

Para los anfitriones que tienen direcciones alias, una sola interfaz puede tener muchas direcciones IP. Por ejemplo, podemos asignar eth0:0 a 192.168.1.16, eth0:1 a 192.168.1.17 y eth0:2 a 192.168.1.18. En este caso, si la conexión sale de la interfaz eth0 y la aplicación no se ligó a una interfaz específica, Linux siempre elegirá la dirección IP no alias; es decir la 192.168.1.15 para eth0. Si la aplicación eligió ligarse a una dirección IP, digamos la 192.168.1.17, Linux usará esa dirección IP como la IP fuente, sin importar si la conexión sale de eth0 o de eth1.

RESUMEN

En este capítulo hemos visto cómo se usan los comandos **ifconfig** y **route** para configurar las direcciones IP y las entradas de rutas, respectivamente. También hemos visto cómo usar estos comandos juntos para estructurar un router Linux sencillo.

Aun cuando, con anterioridad en el libro, cubrimos los módulos del núcleo, los trajimos a colación de nuevo en el contexto específico de los controladores de redes. Recuerde que las interfaces de redes no siguen el mismo método de acceso que el de la mayor parte de los dispositivos con una entrada /dev.

Por último, recuerde que al hacer cambios de dirección IP y de encaminamiento, asegúrese de agregar cualesquiera cambios y todos ellos a los scripts de arranque. Puede ser que quiera programar un reinicio si se encuentra en un sistema de producción, para asegurarse de que los cambios funcionan como se esperaba, de modo que, más tarde, no lo agarren con la guardia abajo.

Si tiene interés en obtener más detalles acerca del encaminamiento, vale la pena mirar con un mayor cuidado el capítulo siguiente y algunas de las características avanzadas de encaminamiento de Linux. Éste ofrece un conjunto rico de funciones que, aun cuando en general no se usan en los entornos de servidores, pueden conducir a un sistema poderoso de encaminamiento. Para quienquiera que se encuentre interesado en el encaminamiento dinámico con el uso de RIP, OSPF o BGP, asegúrese de echar una mirada a Zebra (<http://www.zebra.org>). Con Zebra puede ejecutar un sistema de encaminamiento dinámico intensamente configurable que puede compartir actualizaciones de rutas con cualquier router estándar, incluyendo el equipo Cisco grande.

CAPÍTULO 13



Configuración del firewall de Linux

En lo que se antoja como hace mucho, mucho tiempo, la Internet era un lugar muy amigable. Los usuarios de la red tenían investigaciones que hacer y, como consecuencia, tenían mejores cosas que hacer que desperdiciar su tiempo fisgando la infraestructura de otras personas. La seguridad del área estaba en su lugar y, en gran parte, era para evitar que los bromistas prácticos hicieran cosas tontas. Muchos administradores no realizaron un esfuerzo serio para asegurar sus sistemas, dejando a menudo en su lugar las contraseñas predeterminadas del administrador.

Por desgracia, conforme creció la población, del mismo modo lo hizo la amenaza de los aburridos y los maliciosos. La necesidad de poner barreras entre la Internet y las redes privadas empezó a volverse cada vez más un lugar común a principios de la década de 1990. Artículos como "An Evening with Berferd" (Una tarde con Berferd) y "Design of a Secure Internet Gateway" (Diseño de una puerta segura de acceso a la Internet), de Bill Cheswick, significaron la primera idea popular de lo que se ha convertido en un firewall (los dos artículos se encuentran en el sitio Web de Bill, en <http://www.cheswick.com/ches>).

Desde entonces, la tecnología de los firewalls ha pasado por una gran cantidad de cambios.

El firewall y el sistema de filtrado de paquetes de Linux han recorrido también un largo camino con estos cambios; desde una implementación inicial tomada prestada de BSD, pasando por cuatro reescrituras importantes (núcleos 2.0, 2.2, 2.4 y 2.6) y tres interfaces del nivel usuario (ipfwadm, ipchains e iptables). La infraestructura (tanto del núcleo como de las herramientas del usuario) actual del filtro de paquetes y el firewall de Linux se menciona como "Netfilter".

En este capítulo empezamos con una discusión de cómo funciona Linux Netfilter, seguido por la forma como se aplican estos términos en el estuche de herramientas de Linux 2.6 y finalizando con varios ejemplos de configuración.

NOTA ¡No suponemos que por leer este capítulo usted sea un experto en firewall! En este capítulo se da una introducción al sistema Netfilter y a la manera en que funcionan los firewalls con guía suficiente como para hacer segura una red sencilla. Se han escrito volúmenes enteros acerca de cómo funcionan los firewalls, cómo deben configurarse y de los detalles intrincados de cómo deben desplegarse. Si el lector está interesado en la seguridad más allá del alcance de una simple configuración, debe elegir algunos de los libros recomendados al final del capítulo.

CÓMO FUNCIONA EL NETFILTER

El principio que se encuentra detrás de Netfilter es sencillo: proporciona medios sencillos para tomar decisiones acerca de cómo debe fluir un paquete. Para hacer que la configuración sea más fácil, Netfilter suministra una herramienta, llamada **iptables**, que se puede ejecutar desde la línea de comandos. La herramienta **iptables** facilita hacer una lista de las reglas, y agregar y quitar éstas según lo necesite el sistema. Todo el código real que procesa los paquetes según la configuración que usted le dé en realidad se ejecuta en el interior del núcleo.

Para llevar a cabo esto, la infraestructura de Netfilter descompone la tarea en tres tipos de operaciones: *NAT*, *desmenuzadora* y *filtro*. Cada operación tiene su propia tabla de operaciones que se pueden realizar con base en reglas definidas por el administrador. La tabla NAT es responsable de manejar la *Network Address Translation* (Traducción de direcciones de la red); es decir, producir o cambiar direcciones IP para una dirección IP fuente o destino en particular. El uso más común de esto es permitir que múltiples sistemas tengan acceso a otra red (por lo general, la Internet) desde una sola dirección IP. Cuando se combina con el seguimiento del rastro de las conexiones, ésta es la esencia del firewall de Linux.

La tabla desmenuzadora es la responsable de alterar o marcar los paquetes. El número de usos posibles de esta tabla es enorme; sin embargo, también se usa con poca frecuencia. Un ejemplo de su uso sería cambiar los bits ToS (Type of Service, Tipo de servicio) del encabezado TCP de modo que se puedan aplicar los mecanismos de Quality of Service (QoS, Calidad del servicio) a un paquete, más adelante el encaminamiento o en otro sistema.

Por último, la tabla filtro es responsable de proporcionar el filtrado básico de los paquetes. Ésta se puede usar de manera selectiva para permitir o bloquear el tráfico según cualesquiera reglas que usted aplique al sistema. Un ejemplo de filtrado es bloquear todo el tráfico, excepto aquel destinado al puerto 22 (ssh) o el 25 (SMTP).

Un compendio de NAT

La Network Address Translation (NAT) permite a los administradores esconder los anfitriones de ambos lados de un router de modo que los dos puedan, por cualesquiera razones, permanecer dichosos de estar ignorantes de la existencia del otro. Bajo Netfilter, NAT se puede descomponer en tres categorías: Source NAT (SNAT, NAT Fuente), Destination NAT (DNAT, NAT Destino) y Masquerading (Mascarada).

SNAT es responsable de cambiar cuáles son la dirección IP y el puerto fuentes, de modo que un paquete aparezca como proveniente de una IP definida por el administrador. Ésta es la de uso más común en el caso en el que una red privada necesita usar una dirección IP visible desde el exterior. Para usar una SNAT, el administrador debe saber cuál es la nueva dirección IP fuente cuando se está definiendo la regla. En el caso en el que no se conoce (por ejemplo, la dirección IP la define en forma dinámica un ISP), el administrador debe usar Masquerading (que se define en breve). Otro ejemplo del uso de SNAT es cuando un administrador quiere hacer que un anfitrión específico de una red (por lo general, privada) aparezca como otra dirección IP (por lo general, pública). Cuando se ha terminado de aplicar SNAT, necesita aplicarse más tarde en las etapas de procesamiento del paquete, de modo que todas las demás partes de Netfilter vean la dirección IP fuente original, antes de que el paquete salga del sistema.

DNAT es responsable de cambiar la dirección IP y el puerto destinos, de modo que el paquete se redirija hacia otra dirección IP. Esto resulta útil para situaciones en las que los administradores desean esconder los servidores de una red privada (por lo común mencionada como DMZ en la manera de hablar de los firewalls) y aplica direcciones IP externas seleccionadas en una dirección interna para el tráfico entrante. Desde el punto de vista de la administración, hacer que se realice DNAT hace que sea más fácil administrar las políticas, ya que todas las direcciones IP visibles desde el exterior son visibles desde un solo anfitrión (también conocido como *punto de choque*) de la red.

Por último, *Masquerading* es sencillamente un caso especial de SNAT. Esto resulta útil en situaciones en las que existen múltiples sistemas, en el interior de una red privada, que necesitan compartir una sola dirección IP, asignada en forma dinámica, hacia el mundo exterior y constituye el uso más común de los firewalls basados en Linux. En un caso de ese tipo, Masquerading hará que todos los paquetes aparezcan como si se hubieran originado desde la dirección IP del dispositivo NAT, escondiendo de este modo la estructura de la red privada de usted. El uso de este método de NAT también le permite a la red privada de usted usar los espacios IP privados de RFC 1918, como se hace ver en el capítulo 11 (192.168.0.0/16, 172.16.0.0/12 y 10.0.0.0/8).

Ejemplos de NAT

En la figura 13-1 se muestra un ejemplo sencillo en donde un anfitrión (192.168.1.2) está intentando conectarse a un servidor (200.1.1.1). Si, en este caso, se usara SNAT o Masquerading, se aplica-

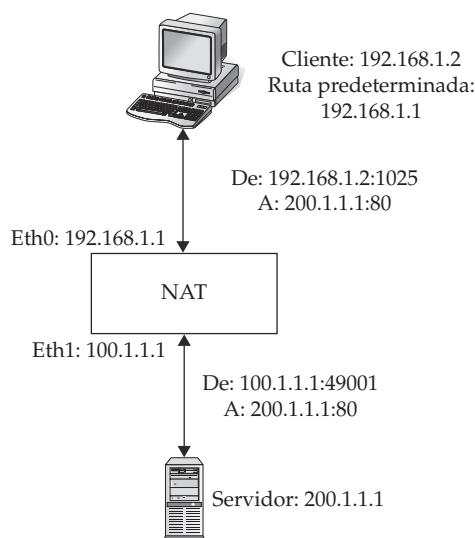


Figura 13-1. Uso de SNAT en una conexión

ría una transformación al paquete de modo que se cambie la dirección IP fuente por la dirección IP externa de NAT (100.1.1.1). Desde el punto de vista del servidor, se está comunicando con el dispositivo NAT, no en forma directa con el anfitrión. Desde el punto de vista del anfitrión, tiene acceso sin obstrucciones hacia la Internet pública. Si hubiera múltiples clientes detrás del dispositivo NAT (digamos, el 192.168.1.3 y el 192.168.1.4), la NAT transformaría todos sus paquetes de modo que aparecieran como si se originaran también desde la 100.1.1.1.

¡Qué pena! Esto trae un pequeño problema. El servidor va a enviar algunos paquetes de regreso; pero cómo el dispositivo NAT va a saber a quién enviar cuál paquete? En esto se encuentra la magia: el dispositivo NAT mantiene una lista interna de las conexiones de los clientes y de las conexiones del servidor asociado, llamadas *flujos*. Como consecuencia, en el primer ejemplo la NAT está manteniendo un registro de que “192.168.1.1:1025 se convierte en 100.1.1.1:49001, el cual está comunicándose con el 200.1.1.1:80”. Cuando 200.1.1.1:80 envía un paquete de regreso a 100.1.1.1:49001, el dispositivo NAT, en forma automática, altera el paquete de modo que la dirección IP destino se fija en 192.168.1.1:1025 y, a continuación, lo pasa al cliente en la red privada.

En su forma más sencilla, un dispositivo NAT sólo está siguiendo el rastro a los flujos. Cada flujo se mantiene abierto mientras vea tráfico. Si la NAT no ve tráfico en un flujo dado, durante algún periodo, el flujo se elimina automáticamente. Estos flujos no tienen idea acerca del contenido de la propia conexión, sólo que está pasando tráfico entre dos puntos extremos y el trabajo de NAT es garantizar que los paquetes lleguen como lo espera cada punto extremo.

Miremos ahora el caso inverso, como se muestra en la figura 13-2. Un cliente de la Internet quiere conectarse a un servidor de una red privada, a través de una NAT. Con el uso de DNAT en esta situación podemos hacer responsabilidad de la NAT que acepte los paquetes en beneficio del servidor, transforme la IP destino de esos paquetes y, a continuación, los entregue al servidor.

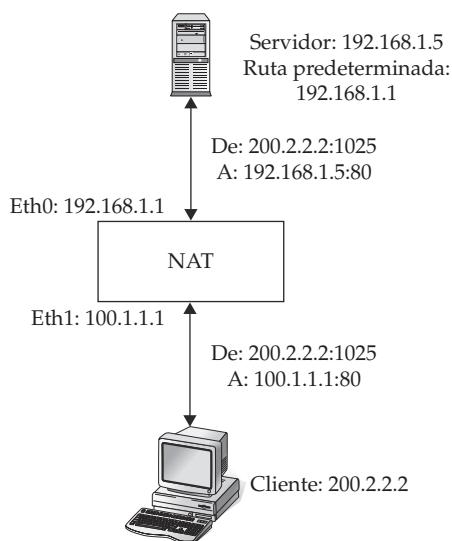


Figura 13-2. Uso de DNAT en una conexión

Cuando el servidor regresa paquetes al cliente, la NAT debe mirar el flujo asociado y cambiar la dirección IP fuente del paquete de modo que se lea proveniente del dispositivo NAT, en lugar que del propio servidor. Volviendo esto hacia las direcciones IP mostradas en la figura 13-2, vemos un servidor en 192.168.1.5:80 y un cliente en 200.2.2.2:1025. El cliente se conecta a la dirección IP de la NAT, 100.1.1.1:80, y esa NAT transforma el paquete de modo que la dirección destino IP sea 192.168.1.5. Cuando el servidor envía un paquete de regreso, el dispositivo NAT realiza lo inverso, de modo que el cliente piense que está hablando con 100.1.1.1. [Note que esta forma particular de NAT también se menciona como Port Address Translation (Traducción de dirección del puerto) o PAT.]

Seguimiento del rastro de la conexión y NAT

Aun cuando NAT parece ser una gran manera de proporcionar seguridad en la superficie, por desgracia no es suficiente. El problema con NAT es que no entiende el contenido de los flujos y si debe bloquearse un paquete porque es una violación del protocolo. Por ejemplo, supongamos que tenemos estructurada una red como se muestra en la figura 13-2. Cuando llega una nueva conexión para el servidor Web, sabemos que debe ser un paquete TCP SYN. No existe otro paquete válido para la finalidad de establecer una nueva conexión. Sin embargo, con una NAT ciega, el paquete será reenviado sin importar si es un TCP SYN o no.

Para hacer que NAT sea más útil, Linux ofrece el *seguimiento del rastro de la conexión de estado completo*. Esta característica le permite a NAT examinar de manera inteligente el encabezado de un paquete y determinar si tiene sentido desde el nivel de un protocolo TCP. De este modo, si llega un paquete para una nueva conexión TCP que no es un TCP SYN, el seguimiento del rastro de la conexión de estado completo lo rechazará, sin poner en riesgo al propio servidor. Incluso mejor, si se establece una conexión válida y una persona maliciosa intenta embromar metiendo

un paquete aleatorio en el flujo, el seguimiento del rastro de la conexión de estado completo dejará caer el paquete, a menos que se ajuste a todos los criterios para que sea un paquete válido entre los dos puntos extremos (una hazaña muy difícil, a menos que el atacante sea capaz de olfatear el tráfico adelante en el tiempo).

Conforme discutamos NAT en todo el resto de este capítulo, tenga presente que siempre que pueda tenerse NAT, se puede tener seguimiento del rastro de la conexión de estado completo.

Protocolos amigables para NAT

A medida que cubrimos NAT con detalle más profundo, es posible que el lector haya advertido que parece que siempre estamos hablando acerca de conexiones únicas recorriendo la red. Para los protocolos que sólo necesitan una conexión para trabajar (como HTTP) y para aquellos que no se apoyan en la comunicación con la dirección IP real del cliente o del servidor (como SMTP), esto es magnífico. ¿Pero qué sucede cuando usted sí tiene un protocolo que necesita conexiones múltiples o hacer pasar las direcciones IP reales? Bien, tenemos un problema.

Existen dos soluciones para manejar estos protocolos: usar una NAT que conozca de aplicaciones o un proxy de plena aplicación. En el primer caso, en general NAT realizará el menor trabajo posible para hacer que el protocolo lo recorra de manera correcta, como componer direcciones IP a la mitad de una conexión y agrupar lógicamente conexiones múltiples porque están relacionadas entre sí. FIP NAT es un ejemplo de las dos situaciones: la NAT debe alterar una corriente FTP activa de modo que la dirección IP que está incrustada se componga para mostrar la dirección IP de la propia NAT, y éste sabrá esperar una conexión de regreso del servidor y saber redirigirla de regreso al cliente apropiado.

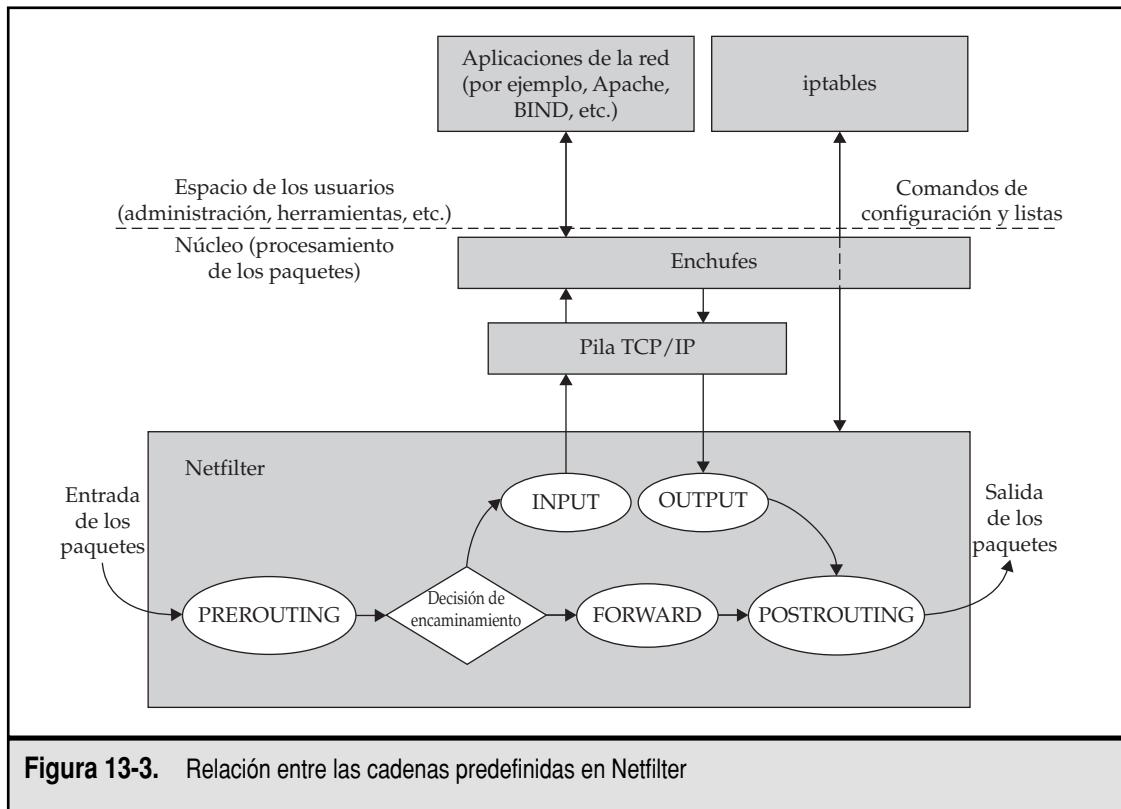
Para protocolos más complejos o aquellos en donde se necesita el pleno conocimiento de la aplicación para asegurarlos en forma correcta, lo típico es que se requiere un proxy del nivel de aplicación. El proxy de aplicaciones tendría el trabajo de terminar la conexión desde la red interior e iniciarla, en beneficio del cliente, en la red exterior. Cualquier tráfico de retorno tendría que recorrer el proxy antes de ir de regreso al cliente.

Desde un punto de vista práctico, existen muy pocos protocolos que en realidad necesitan recorrer una NAT y lo normal es que estos protocolos ya sean amigables para NAT, en el sentido de que sólo requieren una sola conexión cliente a servidor. FTP activo es el único protocolo de uso frecuente que necesita un módulo especial en Netfilter. Un número cada vez mayor de protocolos complejos están ofreciendo reservas sencillas, amigables para NAT, que los hacen más fáciles de desplegar. Por ejemplo, la mayor parte de las aplicaciones de mensajero instantáneo, medios continuos y telefonía IP están ofreciendo reservas amigables para NAT.

Conforme cubramos diferentes configuraciones de Netfilter, introduciremos algunos de los módulos que soportan estos protocolos.

Cadenas

Para cada tabla existe una serie de *cadenas* por las que pasa un paquete. Una cadena es sólo una lista de reglas que actúan sobre un paquete que fluye por el sistema. En Netfilter hay cinco cadenas predefinidas: PREROUTING, FORWARD, POSTROUTING, INPUT y OUTPUT (preencaminamiento, reenvío, posencaminamiento, entrada y salida). En la figura 13-3 se muestra su relación



entre sí. Sin embargo, el lector debe advertir que la relación entre TCP/IP y Netfilter, como se muestra en la figura, es lógica.

Cada una de las cadenas predefinidas puede llamar reglas que se encuentran en una de las tablas predefinidas (NAT, desmenuzadora o filtro). No todas las cadenas pueden llamar cualquier regla en cualquier tabla; cada cadena sólo puede llamar reglas en una lista definida de tablas. En las secciones que siguen, cuando expliquemos qué hace cada una de las cadenas, discutiremos cuáles tablas se pueden usar desde cada cadena.

Si lo desean, los administradores pueden agregar más cadenas al sistema. Un paquete que se ajusta a una regla entonces puede, a su vez, llamar otra cadena de reglas definidas por el administrador. Esto facilita repetir una lista de múltiples reglas, desde cadenas diferentes. Más adelante, en este capítulo, veremos ejemplos de esta clase de configuración.

Todas las cadenas predeterminadas son un miembro de la tabla desmenuzadora. Esto quiere decir que, en cualquier punto a lo largo del trayecto, es posible marcar o alterar el paquete de una manera arbitraria. Sin embargo, la relación entre las otras tablas y cada cadena varía por cadena. En la figura 13-4 se puede ver una representación visual de todas las relaciones.

Recorramos cada una de estas cadenas para comprender estas relaciones.

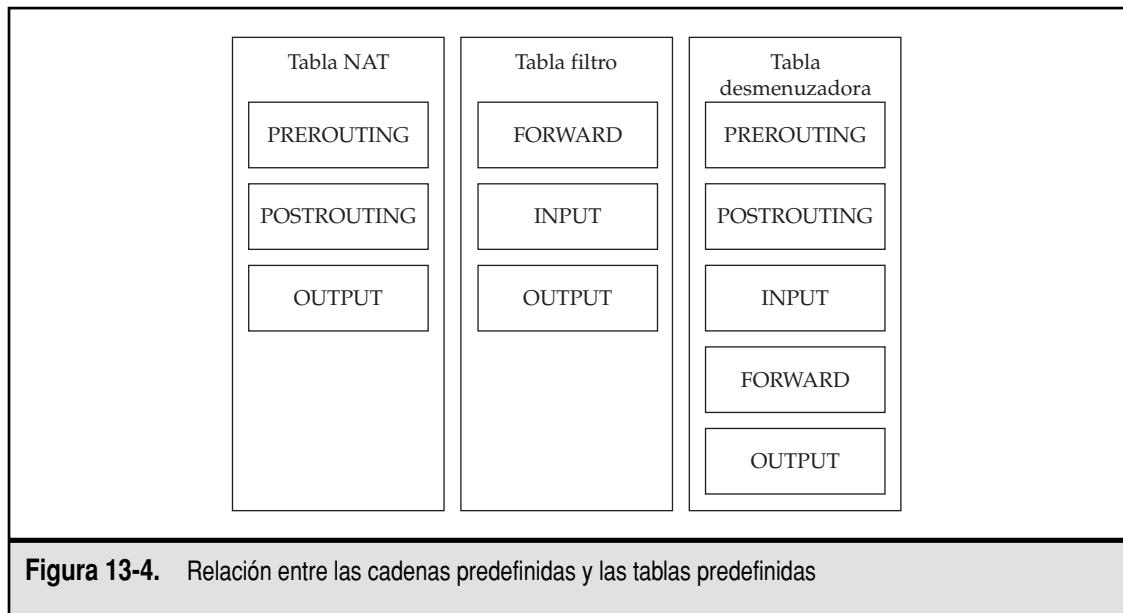


Figura 13-4. Relación entre las cadenas predefinidas y las tablas predefinidas

PREROUTING

La cadena PREROUTING es la primera contra la que choca un paquete al entrar al sistema. Esta cadena puede llamar reglas que se encuentran en una de dos tablas: NAT y desmenuzadora. Desde una perspectiva de NAT, éste es el punto ideal en el cual realizar un Destination NAT (DNAT), con lo cual se cambia la dirección IP destino de un paquete.

Los administradores que consideran seguir el rastro de las conexiones para los fines de un firewall deben iniciar aquí ese seguimiento, ya que es importante seguir el rastro de las direcciones IP originales junto con cualquier dirección NAT proveniente de una operación DNAT.

FORWARD

Se llama la cadena FORWARD sólo en el caso en el que se activa el reenvío IP y el paquete está destinado para un sistema que no sea el del propio anfitrión. Si, por ejemplo, el sistema Linux tiene la dirección IP 172.16.1.1 y se configura para encaminar paquetes entre la Internet y la red 172.16.1.0/24, y un paquete proveniente de 1.1.1.1 está destinado a 172.16.1.10, este paquete recorrerá la cadena FORWARD.

La cadena FORWARD llama reglas que se encuentran en las tablas filtro y desmenuzadora. Esto significa que, en este punto, el administrador puede definir reglas de filtrado de paquetes que se aplicarán a cualesquiera de éstos que van hacia la red en la ruta o vienen de ella.

INPUT

Sólo se llama la cadena INPUT cuando un paquete está destinado para el propio anfitrión. Las reglas que se ejecutan contra un paquete se terminan antes que éste vaya hacia arriba de la pila y

llegue a la aplicación. Por ejemplo, si el sistema Linux tiene la dirección IP 172.16.1.1, el paquete tiene que estar destinado a esta dirección para que se aplique cualquiera de las reglas que se encuentran en la cadena INPUT. Si una regla deja caer todos los paquetes destinados al puerto 80, cualquier aplicación que escuche conexiones por ese puerto nunca verá alguna.

La cadena INPUT llama las reglas de las tablas filtro y desmenuzadora.

OUTPUT

Se llama la cadena OUTPUT cuando los paquetes se envían desde aplicaciones que se están ejecutando en el propio anfitrión. Por ejemplo, si un administrador en la CLI intenta usar ssh para conectarse a un sistema remoto, la cadena OUTPUT verá el primer paquete de la conexión. Los paquetes que regresan del anfitrión remoto entrarán por PREROUTING e INPUT.

Además de las tablas filtro y desmenuzadora, la cadena OUTPUT puede llamar reglas que estén en la tabla NAT. Esto permite a los administradores configurar transformaciones NAT para que se lleven a cabo sobre los paquetes salientes que se generen desde el propio anfitrión. Aun cuando esto es atípico, la característica hace que en realidad los administradores realicen operaciones NAT del estilo PREROUTING sobre los paquetes (recuerde, si el paquete se origina desde el anfitrión, nunca tiene posibilidad de pasar por la cadena PREROUTING).

POSTROUTING

La cadena POSTROUTING puede llamar las tablas NAT y desmenuzadora. En esta cadena, los administradores pueden alterar la dirección IP fuente para los fines de Source NAT (SNAT). Éste también es otro punto en el que se puede tener seguimiento del rastro de la conexión para los fines de estructuración de un firewall.

INSTALACIÓN DE NETFILTER

Las buenas noticias son que si tiene una distribución moderna de Linux, es probable que ya tenga instalado, compilado y trabajando Netfilter. Una comprobación rápida es sencillamente intentar ejecutar el comando **iptables** de este modo:

```
[root@hostA:/root]# iptables -L
```

Note que algunas distribuciones no incluyen el directorio **/sbin** en la trayectoria y existe una buena posibilidad de que el programa **iptables** viva allí. Si no está seguro, intente usar una de las siguientes trayectorias completas: **/sbin/iptables**, **/usr/sbin/iptables**, **/usr/local/bin/iptables**, o bien, **/usr/local/sbin/iptables**. Los directorios **/bin** y **/sbin** ya deben estar en su trayectoria y deben haber sido verificados cuando intentó **iptables** sin una trayectoria absoluta.

Si el comando le dio una lista de cadenas y tablas, entonces usted ya tiene Netfilter instalado. De hecho, ¡hay una buena posibilidad de que el proceso de instalación ya activó algunos filtros! Fedora Core, por ejemplo, da una opción de configurar un sistema de baja, media o alta seguridad en el momento de la instalación. Esa preferencia definió cuál configuración de Netfilter usar.

Con Netfilter ya presente, puede saltarse esta sección, si no tiene interés en cómo instalar Netfilter a mano.

SUGERENCIA Como administrador, nunca sabe cuándo se encontrará encarado con la instalación de Netfilter a mano. Tomar una pequeña parte de esta sección todavía puede ser una buena idea, incluso si no necesita instalarlo precisamente ahora.

El proceso completo de instalación de Netfilter en realidad se divide en dos partes: activación de las características en el curso del proceso de compilación del núcleo y la compilación de las herramientas de administración. Empecemos con lo primero.

Activación de Netfilter en el núcleo

La mayor parte del código de Netfilter en realidad vive en el interior del núcleo y se embarca con la distribución estándar de kernel.org de Linux. Para activar Netfilter, sólo necesita activar las opciones correctas en el transcurso del paso de configuración del núcleo de compilación de un núcleo. Si no está familiarizado con el proceso de compilación de un núcleo, vea el capítulo 9 para obtener detalles.

Sin embargo, Netfilter tiene una gran cantidad de opciones. En esta sección cubriremos cuáles son esas opciones y cuáles quiere usted seleccionar.

Opciones requeridas del núcleo

Existen tres módulos requeridos que deben soportarse: Network Packet Filtering (Filtrado de paquetes de la red), IP Tables (Tablas IP) y Connection Tracking (Seguimiento del rastro de las conexiones).

El primero se encuentra bajo el menú Networking Support | Networking Options (Soporte de la operación en red | Opciones de la operación en red) al compilar el núcleo. Esto proporciona la funcionalidad básica del armazón de Netfilter en el núcleo. Sin esta opción activada, ninguna de las otras opciones dadas en la lista funcionará. Note que esta característica no se puede compilar como un módulo del núcleo; está dentro o fuera.

El segundo, IP Tables, se encuentra bajo Networking Support | Networking Options | Network Packet Filtering | IP: Netfilter Configuration. La finalidad de este módulo es proporcionar la interfaz de las IP Tables y la administración para el sistema de Netfilter. Técnicamente, este módulo es opcional, ya que es posible usar las antiguas interfaces ipchains o ipfwadm; sin embargo, a menos que tenga una razón específica para adherirse a la vieja interfaz, debe usar por el contrario IP Tables. Si usted se encuentra en el proceso de emigrar de su vieja configuración ipchains/ipfwadm hacia las IP Tables, querrá que todos los módulos estén compilados y disponibles para usted.

Por último, la opción Connection Tracking (la cual se puede hallar en el mismo lugar que la opción IP Tables) ofrece la capacidad de agregar soporte para realizar un inteligente seguimiento del rastro de las conexiones TCP/IP y el soporte específico para protocolos clave como FTP. De modo semejante a la opción IP Tables, esto se puede compilar como un módulo.

Opciones sensibles del núcleo

Con las opciones que acaban de nombrarse compiladas en el núcleo, técnicamente cuenta con lo suficiente reunido para hacer que Netfilter funcione para la mayor parte de las aplicaciones. Sin embargo, hay unas cuantas opciones que hacen que la vida sea más fácil, proporcionan seguridad adicional y soportan algunos protocolos comunes. Para todos los fines prácticos, debe

considerar estas opciones como requisitos. Todas las opciones que siguen se pueden compilar como módulos, de modo que sólo aquellas en uso activo se cargan en la memoria. Hagamos un recorrido por ellas.

- ▼ **FTP Protocol Support** (Soporte del protocolo FTP) Se dispone de esta opción una vez que se selecciona Connection Tracking. Con ella, puede manejar en forma correcta las conexiones FTP activas a través de NAT. Esto es necesario, porque, como se cubrirá en el capítulo 17, FTP activo requiere que se establezca una conexión separada desde el servidor de regreso al cliente cuando se transfieran datos (por ejemplo, listas de directorios, transferencias de archivos, etcétera). De modo predeterminado, NAT no sabrá qué hacer con la conexión iniciada del servidor. Con el módulo FTP se le dará a NAT la inteligencia para manejar en forma correcta el protocolo y asegurarse de que la conexión asociada la haga regresar al cliente apropiado.
- **IRC Protocol Support** (Soporte del protocolo IRC) Se dispone de esta opción una vez que se selecciona Connection Tracking. Si usted espera que los usuarios que se encuentran detrás de NAT quieran usar IRC para comunicarse con otros en la Internet, se requerirá este módulo para manejar de modo correcto la conectividad, las peticiones IDENT y las transferencias de archivos.
- **Connection State Match** (Concordancia del estado de la conexión) Se dispone de esta opción una vez que se activa IP Tables Support. Con ella, el seguimiento del rastro de la conexión gana la funcionalidad de estado completo que se discutió en la sección “Seguimiento del rastro de la conexión y NAT”, con anterioridad en el capítulo. Esto debe considerarse como un requisito para cualquiera que configura su sistema como un firewall.
- **Packet Filtering** Esta opción se requiere si desea proporcionar opciones de filtro de paquetes. Esto debe considerarse como un requisito.
- **REJECT Target Support** (Soporte RECHAZO del destino) Esta opción está relacionada con la opción Packet Filtering en el sentido de que proporciona una manera de rechazar un paquete, con base en el filtro de paquetes, enviando un error ICMP de regreso a la fuente de un paquete, en lugar de sólo dejarlo caer. Dependiendo de su red, esto puede ser útil; sin embargo, si su red está encarando la Internet, la opción REJECT no es una buena idea. Es mejor dejar caer silenciosamente los paquetes que no quiere, en lugar de generar más tráfico.
- **LOG Target Support** (Soporte REGISTRO del destino) Con esta opción puede configurar el sistema para registrar un paquete que se ajuste a una regla. Por ejemplo, si quiere registrar todos los paquetes que se dejan caer, esta opción lo hace posible.
- **Full NAT** (NAT pleno) Esta opción es un requisito para proporcionar funcionalidad NAT en Netfilter. Debe considerar esta opción como un requisito.
- **MASQUERADE Target Support** (Soporte ENMASCARADO del destino) Esta opción es un requisito para proporcionar una manera fácil para esconder una red privada a través de NAT. Este módulo crea internamente una entrada NAT.

- **REDIRECT Target Support (REDIRIGIR)** Esta opción permite al sistema redirigir un paquete al propio anfitrión NAT. Usar esta opción le permite a usted estructurar programas proxy transparentes, los cuales son útiles cuando no es factible configurar cada cliente de su red con los ajustes proxy apropiados, o bien, si la propia aplicación no es propicia para conectar a un servidor proxy.
- **NAT of Local Connections (NAT de conexiones locales)** Esta opción le permite a usted aplicar reglas DNAT a paquetes que se originan desde el propio sistema NAT. Si no está seguro si necesitará esto más tarde, siga adelante y compílelo dentro.
- ▲ **Packet Mangling (Desmenuzado de paquetes)** Esta opción se agrega a la tabla desmenuzadora. Si piensa que querrá la capacidad de manipular o marcar paquetes por separado para opciones como Quality of Service, deseará activar este módulo.

Otras opciones

Existen muchas opciones adicionales que se pueden activar con Netfilter. La mayor parte de ellas se fijan para compilarse como módulos de manera predeterminada, lo cual quiere decir que puede compilarlos ahora y decidir si en realidad quiere usarlos más adelante, sin ocupar memoria preciosa.

A medida que pase por el proceso de compilación, tome algo de tiempo para mirar los otros módulos y leer sus secciones de ayuda. Muchos módulos ofrecen funciones pequeñas interesantes que puede usted hallar prácticas para realizar cosas raras que por lo común no son posibles con los firewalls. En otras palabras, estas funciones le permiten mostrar en realidad el poder de Netfilter y Linux.

Por supuesto, existe una consideración que hacer en relación con lo oscuro. Cuando un módulo no se está usando con intensidad, tampoco se prueba con intensidad. Si está esperando ejecutar esta NAT como un sistema de producción, puede ser que quiera adherirse a lo básico y mantener las cosas simples. Lo sencillo facilita más la detección de sus fallas, su mantenimiento y, por supuesto, es seguro.

Compilación de las tablas IP

Una vez que haya compilado e instalado su núcleo, está listo para instalar la segunda parte de Netfilter: la herramienta **IPTables**. Esta herramienta es responsable de proporcionar una interfaz para crear y administrar las reglas de Netfilter y de detectar las fallas en éstas. Como su interfaz, se estructura como una aplicación que se ejecuta en la CLI y, por tanto, está separada del propio paquete del núcleo.

Empiece por descargar el paquete más reciente de **iptables** de <http://www.netfilter.org/>. En el momento en que se está escribiendo esto, ése es netfilter-1.3.0.

Una vez que tenga el paquete **iptables**, guárdelo y desempáquelo en **/usr/local/src**.

```
[root@hostA src]# tar -xvzf iptables-1.3.0.tar.bz2
```

Esto creará el directorio **iptables-1.3.0**, en el cual se desempacará el código fuente. Muévase hacia el directorio **iptables**.

```
[root@hostA src]# cd iptables-1.3.0
```

Tómese un momento para ver el archivo **INSTALL** que viene con el paquete. Dará una lista de las noticias más recientes acerca del proceso de instalación, cualesquiera advertencias y algunas sugerencias para la detección de fallas, para los problemas comunes. En la mayor parte de los casos, sólo necesitará llamar el comando **make** con un parámetro que especifique la ubicación del núcleo actual.

Para estructurar las **iptables**, ejecute

```
[root@hostA iptables-1.3.0]# make KERNEL_DIR=/usr/src/kernel
```

Si su núcleo se encuentra en una ubicación diferente, cambie el valor **KERNEL_DIR** para reflejar eso.

NOTA “¿Qué hago si no tengo instalado el código fuente del núcleo?” Si está actualizando el comando **iptables** pero está usando el núcleo que vino con su distribución de Linux, es posible que la fuente del núcleo no esté instalada en el sistema (por ejemplo, Fedora Core no instala la fuente del núcleo de modo predeterminado). No se preocupe, **iptables** no está interesado en una copia completa del núcleo actual sino sólo en dónde puede hallar los archivos *incluir* (include) de ese núcleo actual. Estos archivos los usan los desarrolladores para compartir bits comunes del código. Para darle la vuelta a eso, sencillamente cambie el ajuste **KERNEL_DIR** a **/usr/include**.

Una vez que tenga estructurada la herramienta **iptables**, instálela y ¡ha terminado!

```
[root@hostA iptables-1.3.0]# make install
```

CONFIGURACIÓN DE NETFILTER

Existe una buena posibilidad de que su instalación de Linux ya tenga configurados algunos ajustes de Netfilter para usted, en especial si está usando una distribución más o menos reciente. Los escritorios gráficos que arrancan en forma automática pueden ya haber permitido que usted juegue con algunos de los ajustes de Netfilter en la forma de políticas de seguridad para la conectividad de la red.

Desde un punto de vista administrativo, esto le da tres posibilidades a escoger: adherirse a la GUI para configurar Netfilter, aprender cómo administrar el sistema usando el conjunto existente de scripts o moverse hacia la línea de comandos.

Si elige adherirse a una GUI, es necesario que esté consciente del hecho de que existen múltiples GUI disponibles para Linux, además de la que puede haber sido embarcada con su sistema. Sin embargo, la clave de su decisión es que una vez que la haya tomado, va a querer adherirse a ella. Aun cuando es posible pasar de la GUI a la CLI, no se recomienda, a menos que sepa cómo administrar a mano los archivos de configuración de la GUI.

La administración del sistema usando el conjunto existente de scripts requiere la menor cantidad de cambios, desde el punto de vista de un script de arranque/paro, ya que está usando la estructura existente; sin embargo, también significa llegar a saber cómo se configura la estructura actual y aprender la manera de editar esos archivos.

Por último, para ignorar los scripts existentes y avanzar por sus propios medios necesita partir de nada, pero tendrá el beneficio de saber con exactitud cómo funciona, cuándo arranca y cómo administrarlo. La desventaja es que también necesitará crear toda la infraestructura de

arranque y detención. Debido a la importancia de la funcionalidad del firewall, no es aceptable agregar de manera sencilla la configuración al final del script `/etc/rc.d/rc.local`, ya que se ejecuta precisamente al final del arranque. Debido al tiempo necesario para la inicialización, la ventana entre el arranque de un servicio y el arranque del firewall ofrece demasiado tiempo como para que potencialmente suceda un ataque.

Guardado de su configuración de Netfilter

Al final de este capítulo tendrá alguna mezcla de reglas definidas con los comandos `iptables`, posiblemente un ajuste en el sistema de archivos `/proc` y la necesidad de cargar módulos adicionales del núcleo en el momento de la inicialización. Para hacer que estos cambios persistan a través de múltiples reinicios, necesitará guardar cada uno de estos componentes, de modo que arranquen como usted lo espera de ellos en el momento de la inicialización.

El guardado bajo Fedora Core y Red Hat es bastante directo. Sencillamente siga estos pasos:

1. Guarde sus reglas de Netfilter con el uso del comando siguiente:

```
[root@hostA ~]# /etc/rc.d/init.d/iptables save
```

2. Agregue los módulos apropiados a la variable `IPTABLES_MODULES` del archivo `/etc/sysconfig/iptables-config`. Por ejemplo, para agregar `ip_conntrack_ftp` e `ip_nat_ftp`, haga que la línea `IPTABLES_MODULES` se lea como sigue:

```
IPTABLES_MODULES="ip_conntrack_ftp ip_nat_ftp"
```

3. Agregue cualesquier cambios al sistema de archivos `proc`, al archivo `/etc/rc.d/rc.local`. Por ejemplo, para activar el reenvío de IP necesitaría agregar la línea siguiente:

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

Para otras distribuciones, los métodos variarán. Si no está seguro acerca de cómo funciona su distribución o está probando ser un mayor dolor de cabeza de lo que vale la pena, sólo desactive los scripts integrados desde la secuencia de arranque y agregue los propios. En el caso de que escriba su propio script, puede usar el siguiente perfil:

```
#!/bin/sh
## Define where iptables and modprobe is located
IPT="/sbin/iptables"
MODPROBE="/sbin/modprobe"

## Add your insmod/depmod lines here.
$MODPROBE ip_tables
$MODPROBE ipt_state
$MODPROBE iptable_filter
$MODPROBE ip_conntrack
$MODPROBE ip_conntrack_ftp
$MODPROBE iptable_nat
$MODPROBE ip_nat_ftp
```

```
## Flush the current chains, remove non-standard chains, and zero counters
$IPT -t filter -F
$IPT -t filter -X
$IPT -t filter -Z
$IPT -t mangle -F
$IPT -t mangle -X
$IPT -t mangle -Z
$IPT -t nat -F
$IPT -t nat -X
$IPT -t nat -Z

## Add your rules here. Here is a sample one to get you started.
$IPT -A INPUT -i lo -j ACCEPT

## Add any /proc settings here.
echo "1" > /proc/sys/net/ipv4/tcp_syncookies
```

El comando **iptables**

El comando **iptables** es la clave para configurar el sistema Netfilter. Una rápida ojeada a su ayuda en línea con el comando **iptables -h** muestra un número impresionante de opciones de configuración. En esta sección recorreremos esas opciones y aprenderemos cómo usarlas.

En el corazón del comando está la capacidad de definir reglas separadas que se hacen parte de una cadena de reglas. Cada regla por separado tiene un criterio de correlación con un paquete y una acción correspondiente. Cuando un paquete recorre un sistema, recorrerá las cadenas apropiadas, como se vio en la figura 13-3 con anterioridad en el capítulo. Dentro de cada cadena, cada regla se ejecutará en orden sobre el paquete. Cuando una regla se ajusta a un paquete, se toma en éste la acción especificada. Estas acciones separadas se mencionan como *objetivos*.

Administración de las cadenas

El formato del comando varía por la acción deseada en la cadena. Éstas son las acciones posibles:

iptables -t table -A chain rule-spec [options]	Agregar rule-spec a chain
iptables -t table -D chain rule-spec	Borre rule-spec de chain
iptables -t table -I chain [rulenumber] rule-spec [options]	Inserte rule-spec al número de regla rulenumber . Si no se especifica número de regla, la regla se inserta en la parte superior de la cadena

<code>iptables -t table -R chain rulenumber rule-spec [options]</code>	Reemplace <code>rulenumber</code> con <code>rule-spec</code> en <code>chain</code>
<code>iptables -t table -L chain [options]</code>	Haga una lista de las reglas en <code>chain</code>
<code>iptables -t table -F chain [options]</code>	Borre (elimine todas) las reglas en <code>chain</code>
<code>iptables -t table -Z chain [options]</code>	Igualle a cero todos los contadores en <code>chain</code>
<code>iptables -t table -N chain</code>	Defina una nueva cadena llamada <code>chain</code>
<code>iptables -t table -X [chain]</code>	Borre <code>chain</code> . Si no se especifica la cadena, se borran todas las cadenas no estándar
<code>iptables -t table -P chain target</code>	Define la política predeterminada para una <code>target</code> cadena. Si ninguna regla se correlaciona para una cadena dada, la política predeterminada envía el paquete a <code>target</code>
<code>iptables -t table -E chain [new-chain]</code>	Renombra <code>chain</code> como <code>new-chain</code>

Recuerde que se tienen tres tablas integradas (NAT, filtro y desmenuzadora) y cinco cadenas integradas (PREROUTING, POSTROUTING, INPUT, FORWARD y OUTPUT). Recuerde que, en la figura 13-4, se muestran sus relaciones.

Sin embargo, a medida que las reglas se vuelven más complejas, a veces es necesario dividirlas en grupos más pequeños. Netfilter le permite hacer esto mediante la definición de su propia cadena y colocarla dentro de la tabla apropiada.

Al recorrer las cadenas estándar, una regla de correlación puede disparar un salto hacia otra cadena en la misma tabla. Por ejemplo, vamos a crear una cadena llamada “to_net10” que maneje todos los paquetes destinados a la red 10.0.0.0/8 que va a través de la cadena FORWARD.

```
[root@hostA ~]# iptables -t filter -N to_net10
[root@hostA ~]# iptables -t filter -A FORWARD -d 10.0.0.0/8 -j to_net10
[root@hostA ~]# iptables -t filter -A to_net10 -j RETURN
```

En este ejemplo la cadena to_net10 no hace sino regresar el control a la cadena FORWARD.

SUGERENCIA Cada cadena debe tener una política predeterminada. Es decir, debe tener una acción predeterminada que tomar en el caso de que un paquete no cumpla con cualquiera de las reglas. Al diseñar un firewall, el procedimiento seguro es fijar que la política predeterminada (usando la opción `-P` en `iptables`) para cada cadena sea DROP (dejar caer) y, a continuación, insertar explícitamente las reglas ALLOW (permitir) para el tráfico de la red que en realidad quiere permitir.

Definición de la rule-spec

En la sección anterior hicimos mención de la *rule-spec*. La rule-spec es la lista de reglas que Netfilter usa para correlacionar un paquete. Si la rule-spec se ajusta a un paquete, Netfilter aplicará la acción deseada sobre él.

Las siguientes son las reglas integradas:

- ▼ **p [!] protocol** Ésta especifica el protocolo IP contra el cual hay que comparar. Usted puede usar cualquier protocolo definido en el archivo **/etc/protocols**, como “tcp”, “udp” o “icmp”. Un valor integrado de “all” indica que se ajustarán todos los paquetes IP. Si el protocolo no está definido en **/etc/protocols**, aquí puede usar el número de protocolo. Por ejemplo, 47 representa “gre”. El signo de admiración (!) niega la verificación. Por tanto **-p ! tcp** significa todos los paquetes que no sean TCP. Si no se proporciona esta opción, Netfilter supondrá “all”. La opción **--protocol** es un alias para esta opción. Un ejemplo de su uso es

```
[root@hostA ~]# iptables -t filter -A INPUT -p tcp --dport 80 -j ACCEPT
```

Esta regla aceptará todos los paquetes destinados al puerto TCP 80 de la cadena INPUT.

- **s [!] address [/mask]** Esta opción especifica la dirección IP fuente contra la cual hay que verificar. Cuando se combina con una netmask opcional, la IP fuente se puede comparar contra un bloque completo de la red. Como con **-p**, el uso del signo de admiración (!) invierte el significado de la regla. De donde, **-s ! 10.13.17.2** significa todos los paquetes que no provengan de 10.13.17.2. Note que la *address* y la *netmask* se pueden abbreviar. Un ejemplo de su uso es

```
[root@hostA ~]# iptables -t filter -A INPUT -s 172.16/16 -j DROP
```

Esta regla dejará caer todos los paquetes provenientes de la red 172.16.0.0/16. Ésta es la misma red que 172.16.0.0/255.255.0.0.

- **d [!] address [/mask]** Esta opción especifica la dirección IP destino contra la cual hay que verificar. Cuando se combina con una netmask opcional, la IP destino se puede comparar contra un bloque completo de la red. Como con **-s**, el uso del signo de admiración (!) niega la regla y la *address* y la *netmask* se pueden abbreviar. Un ejemplo de su uso es

```
[root@hostA ~]# iptables -t filter -A FORWARD -d 10.100.93.0/24 -j ACCEPT
```

Esta regla permitirá todos los paquetes que vayan a través de la cadena FORWARD que estén destinados a la red 10.100.93.0/24.

- ▲ **j target** Esta opción especifica una acción la cual hay que “saltar”. Estas acciones se conocen como *objetivos* (*targets*) en la manera de hablar de **iptables**. Los objetivos que hemos visto hasta ahora han sido ACCEPT, DROP y RETURN. Los dos primeros aceptan y dejan caer los paquetes, respectivamente. El tercero se relaciona con la creación de cadenas adicionales.

Como vimos en la sección anterior, es posible que usted cree sus propias cadenas para ayudarse a mantener las cosas organizadas y para dar lugar a reglas más complejas. Si **iptables** está evaluando un conjunto de reglas en una cadena que no está integrada, el objetivo RETURN le dirá a **iptables** que regrese a la cadena padre. Usando el ejemplo de `to_net10` antes dado,

cuando **iptables** llega al **-j RETURN**, regresa a procesar la cadena FORWARD de donde salió. Si **iptables** ve la acción RETURN en una de las cadenas integradas, ejecutará la regla predeterminada para esa cadena.

Se pueden cargar objetivos adicionales a través de los módulos de Netfilter. Por ejemplo, se puede cargar el objetivo REJECT con **ipt_REJECT**, el cual dejará caer el paquete y enviará de regreso un paquete de error ICMP al remitente. Otro objetivo útil es **ipt_REDIRECT**, el cual puede hacer que se destine un paquete al propio anfitrión NAT, incluso si ese paquete está destinado a alguna otra parte.

- ▼ **i interface** Esta opción especifica el nombre de la interfaz en la cual se recibió un paquete. Ésta resulta práctica para casos en donde deben aplicarse reglas especiales, si un paquete llega de una ubicación física, como una interfaz DMZ. Por ejemplo, si eth1 es su interfaz DMZ y quiere permitirle enviar paquetes al anfitrión en 10.4.3.2, puede usar

```
[root@hostA ~]# iptables -t filter -A FORWARD -i eth1 -d 10.4.3.2 -j ACCEPT
```

- **o interface** Esta opción especifica el nombre de la interfaz en la cual un paquete dejará el sistema. Por ejemplo,

```
[root@hostA ~]# iptables -t filter -A FORWARD -i eth0 -o eth1 -j ACCEPT
```

En este ejemplo se aceptan cualesquiera paquetes que entran desde eth0 y salen hacia eth1.

- **[!] -f** Esta opción especifica si un paquete es un fragmento IP o no. El signo de admiración niega esta regla. Por ejemplo,

```
[root@hostA ~]# iptables -t filter -A INPUT -f -j DROP
```

En este ejemplo, cualesquiera fragmentos que entran en la cadena INPUT se dejan caer automáticamente. La misma regla con lógica negativa sería

```
[root@hostA ~]# iptables -t filter -A INPUT ! -f -j ACCEPT
```

- **c PKTS BYTES** Esta opción le permite a usted fijar los valores de los contadores para una regla en particular cuando se inserta o se anexa a una regla, o reemplaza a ésta, en una cadena. Los contadores corresponden al número de paquetes y de bytes que han recorrido la regla, respectivamente. Para la mayoría de los administradores, ésta es una necesidad rara. Un ejemplo de su uso es

```
[root@hostA ~]# iptables -t filter -I FORWARD -f -j ACCEPT -c 10 10
```

En este ejemplo, una nueva regla que permite los fragmentos de paquetes se inserta en la cadena FORWARD y los contadores de paquetes se fijan en 10 paquetes y 10 bytes.

- **v** Esta opción presentará cualquier salida de **iptables** (por lo común especificada con **-L**) para mostrar datos adicionales. Por ejemplo,

```
[root@hostA ~]# iptables -L -v
```

- **n** Esta opción presentará cualesquier nombres de anfitriones o nombres de puertos en su forma numérica. Normalmente, las **iptables** realizarán la resolución DNS para usted y mostrarán los nombres de los anfitriones, en lugar de las direcciones IP, y los nombres de los protocolos (como smtp), en lugar de los números de puertos (25). Si su sistema DNS no se está ejecutando o si usted no quiere generar cualesquier paquetes adicionales, ésta es una opción útil.

Un ejemplo de esto es

```
[root@hostA ~]# iptables -L -n
```

- **x** Esta opción mostrará los valores exactos de un contador. Normalmente, las **iptables** imprimirán los valores en términos “amigables para las personas” y, por consiguiente, realizan redondeo en el proceso. Por ejemplo, en lugar de mostrar “10310”, las **iptables** mostrarán “10k”.

Un ejemplo de esto es

```
[root@hostA ~]# iptables -L -x
```

- ▲ **line-numbers** Esta opción presentará los números de línea contiguos a cada regla en una cadena. Esto es útil cuando usted necesita insertar una regla a la mitad de una cadena y necesita una lista rápida de las reglas y sus números correspondientes de regla.

Un ejemplo de esto es

```
[root@hostA ~]# iptables -L --line-numbers
```

Extensiones rule-spec con Match

Uno de los aspectos más poderosos de Netfilter es el hecho de que ofrece un diseño “enchufable”. Para los diseñadores, esto significa que es posible hacer extensiones a Netfilter usando una API, en lugar de tener que sumergirse profundamente en el núcleo y realizar con ahínco labores de hacker. Para los usuarios de Netfilter esto quiere decir que se dispone de una amplia variedad de extensiones, más allá del conjunto de características básicas.

Estas extensiones se realizan con la característica Match en la herramienta de la línea de comandos de **iptables**. Mediante la especificación del nombre del módulo deseado, después del parámetro **-m**, **iptables** se encargará de cargar los módulos necesarios del núcleo y, enseguida, ofrecerá un conjunto extendido de parámetros de la línea de comandos. Estos parámetros se usan para ofrecer características más ricas de correlación de los paquetes.

En esta sección discutimos el uso de unas cuantas de estas extensiones que, en el momento en que se está escribiendo este libro, se han probado suficientemente bien como para que se incluyan de manera común con las distribuciones de Linux.

SUGERENCIA Con el fin de obtener ayuda para una extensión match, sencillamente especifique el nombre de la extensión después del parámetro **-m** y, a continuación, dé el parámetro **-h**. Por ejemplo, para obtener ayuda para el módulo ICMP, use

```
[root@hostA ~]# iptables -m icmp -h
```

icmp Este módulo proporciona un parámetro adicional match para el protocolo ICMP:

```
icmp-type [!] typename
```

en donde **typename** es el nombre o número del tipo de mensaje ICMP. Por ejemplo, para bloquear un paquete ping, use lo siguiente:

```
[root@hostA ~]# iptables -t filter -A INPUT -m icmp --icmp-type echo-request
```

Para obtener una lista completa de los tipos de paquetes ICMP soportados, vea la página de ayuda del módulo con la opción **-h**.

limit Este módulo proporciona un método para limitar la rapidez del tráfico de paquetes. Se correlacionará en tanto la rapidez del tráfico esté debajo del límite. Una opción “burst” (andanada) se hace corresponder contra un pico momentáneo en el tráfico pero suspenderá la correlación si el pico se sostiene. Los dos parámetros son

- ▼ **limit rate**
- ▲ **limit-burst number**

La **rate** es el conteo sostenido de paquetes por segundo. El **number** en el segundo parámetro especifica cuántos paquetes aceptar espalda contra espalda en un pico. El valor predeterminado para **number** es 5. El lector puede aceptar esta característica como un simple procedimiento para hacer más lenta una inundación SYN:

```
[root@hostA ~]# iptables -t filter -N syn-flood
[root@hostA ~]# iptables -t filter -A INPUT -p tcp --syn -j syn-flood
[root@hostA ~]# iptables -t filter -A syn-flood -m limit --limit 1/s -j RETURN
[root@hostA ~]# iptables -t filter -A syn-flood -j DROP
```

Con esto se limitará la rapidez de las conexiones a 1 por segundo, con una andanada de hasta 5 conexiones. Esto no es perfecto y una inundación SYN todavía podría negar el acceso a usuarios legítimos con este método; sin embargo, evitará que su servidor describa una espiral hacia la salida de control.

state Este módulo le permite determinar el estado de una conexión TCP a través de los ojos del módulo de seguimiento del rastro de las conexiones. Proporciona una opción adicional,

state state

en donde **state** es INVALID, ESTABLISHED, NEW o RELATED (inválida, establecida, nueva, relacionada). Un estado es INVALID si el paquete en cuestión no se puede asociar con un flujo existente. Si el paquete es parte de una conexión existente, el estado es ESTABLISHED. Si el paquete está iniciando un flujo nuevo, se considera NEW. Por último, si un paquete se asocia a una conexión existente (por ejemplo, una transferencia FTP de datos), entonces es RELATED.

Usando esta característica para asegurarse de que las conexiones nuevas sólo tienen el bit TCP SYN fijado, hacemos lo siguiente:

```
[root@hostA ~]# iptables -t filter -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
```

Al leer este ejemplo, vemos que para un paquete en la cadena INPUT que es TCP y no tiene la bandera SYN fijada, y el estado de una conexión es NEW, dejamos caer el paquete (recuerde que las conexiones TCP deben empezar con un paquete que tenga fijado el bit SYN).

tcp Este módulo nos permite examinar múltiples aspectos de los paquetes TCP. Ya hemos visto algunas de estas opciones (como **--syn**). Enseguida se tiene una lista completa de opciones:

- ▼ **source-port [!] port: [port]** Con esta opción se examina el puerto fuente de un paquete TCP. Si se especifican dos puntos seguidos por un segundo número de puerto, se verifica un rango de puertos. Por ejemplo, “6000:6010” quiere decir “todos los puertos entre 6000 y 6010, inclusive”. El signo de admiración niega este ajuste. Por ejem-

plo, **--source-port ! 25** significa “todos los puertos que no sean el 25”. Un alias para esta opción es **--sport**.

- **destination-port [!] port:[port]** A semejanza de la opción **--source-port**, con ésta se examina el puerto destino de un paquete TCP. Se soportan los rangos y la negación. Por ejemplo, **-destination-port ! 9000:9010** quiere decir “todos los puertos que no están entre 9000 y 9010, inclusive”. Un alias para esta opción es **--dport**.
- **tcp-flags [!] mask comp** Con ésta se verifica que las banderas TCP están fijadas en un paquete. La **mask** le dice a la opción cuáles banderas verificar, y el parámetro **comp** le dice cuáles banderas deben estar fijadas. Tanto **mask** como **comp** pueden ser una lista de banderas separadas por comas. Las banderas válidas son: SYN, ACK, FIN, RST, URG, PSH, ALL, NONE, en donde ALL quiere decir todas las banderas y NONE, ninguna de ellas. El signo de admiración niega el ajuste. Por ejemplo, usar **--tcp-flags ALL SYN,ACK** significa que la opción debe verificar todas las banderas y sólo deben estar fijadas SYN y ACK.
- ▲ **[!] --syn** Con ésta se verifica si está activada la bandera SYN. Es lógicamente equivalente a **--tcp-flags SYN,RST,ACK SYN**. El signo de admiración niega el ajuste.

Un ejemplo usando este módulo comprueba si una conexión hacia el puerto DNS 53 se origina desde el puerto 53, no tiene fijado el bit SYN y tiene fijado el bit URG, en cuyo caso debe dejarlo caer. Note que DNS pasará automáticamente a TCP cuando una petición sea mayor que 512 bytes.

```
[root@hostA ~]# iptables -t filter -A INPUT -p tcp --sport 53 --dport 53 ! --syn
--tcp-flags URG URG -j DROP
```

tcpmss Ésta correlaciona un paquete TCP con Maximum Segment Size (MSS, Tamaño máximo de segmento). El límite legal más bajo para IP es de 576, y el valor más alto es de 1500. La meta en la fijación de un valor MSS para una conexión es evitar la segmentación de paquetes entre dos puntos extremos. Las conexiones por teléfono tienden a usar ajustes de MSS de 576 bytes, en tanto que los usuarios que vienen de enlaces de alta velocidad tienden a usar valores de 1 500 bytes. La opción de la línea de comandos para este ajuste es

mss value:[value]

en donde **value** es el valor MSS contra el cual comparar. Si se dan dos puntos seguidos por un segundo valor, se verifica un rango completo. Por ejemplo,

```
[root@hostA ~]# iptables -t filter -I INPUT -p tcp -m tcpmss --mss 576 -j ACCEPT
[root@hostA ~]# iptables -t filter -I INPUT -p tcp -m tcpmss ! -mss 576 -j ACCEPT
```

Esto proporcionará una manera sencilla de contar cuántos paquetes (y cuántos bytes) están vieniendo de las conexiones que tienen MSS de 576 bytes y cuántas no vienen de ellas. Para ver el estado de los contadores, use **iptables -L -v**.

udp Como el módulo TCP, el UDP proporciona parámetros adicionales para la comprobación de un paquete. Se suministran dos parámetros adicionales:

- ▼ **source-port [!] port:[port]** Con esta opción se comprueba el puerto fuente de un paquete UDP. Si el número del puerto se hace seguir por dos puntos y otro número, se comprueba el rango entre los dos números. Si se usa el signo de admiración, se invierte la lógica.
- ▲ **destination-port [!] port:[port]** A semejanza de la opción **source-port**, con ésta se comprueba el puerto destino UDP.

Por ejemplo,

```
[root@hostA ~]# iptables -t filter -I INPUT -p udp --destination-port 53 -j ACCEPT
```

Este ejemplo aceptará todos los paquetes destinados al puerto 53. Es típico que se fije esta regla para permitir el tráfico hacia los servidores DNS.

SOLUCIONES DEL LIBRO DE RECETAS

De modo que acaba de terminar de leer este capítulo completo y tiene la cabeza girándole un poco. Tantas opciones, tantas cosas que hacer. No se preocupe, eso es para lo que es esta sección: algunas soluciones del libro de recetas para usos comunes del sistema Netfilter de Linux que puede empezar a usar de inmediato mientras las aprende. Por supuesto, si sólo se brincó el capítulo completo y vino hasta aquí, bien, encontrará algunas soluciones del recetario. Sin embargo, vale la pena tomar el tiempo para entender qué están haciendo los comandos, cómo están relacionados y cómo pueden cambiarse. También se volverán unos cuantos ejemplos en posibilidades sin límite.

Con respecto a guardar los ejemplos para usarlos en un sistema de producción, querrá agregar los comandos modprobe a sus scripts de arranque. En Fedora y Red Hat, agregue el nombre del módulo a la variable **IPTABLES_MODULES** en **/etc/sysconfig/iptables-config**. Para otras distribuciones, agregue la línea modprobe completa al archivo **/etc/rc.d/rc.local**. Cualesquier cambios a **/proc** también deben agregarse a **/etc/rc.d/rc.local**. Por último, los usuarios de Fedora y Red Hat pueden guardar sus ajustes a las **iptables** usando el comando que se da enseguida:

```
[root@hostA ~]# /etc/rc.d/init.d/iptables save
```

Para otras distribuciones, edite el script de arranque apropiado.

NAT de tres líneas de Rusty

Rusty Russell, uno de los desarrolladores clave del sistema Netfilter, reconoció que el uso más común para los firewalls de Linux es hacer una red de sistemas disponibles para la Internet vía una dirección IP simple. Ésta es una configuración común en redes de oficina pequeñas y en hogares donde DSL o PPP proporcionan sólo una dirección IP para usar. En esta sección honramos los pasos y la solución de Rusty.

Suponiendo que quiere usar su interfaz ppp0 como su conexión al mundo y que sus otras interfaces (por ejemplo, eth0) se conecten a la red interior, haga lo siguiente:

```
[root@hostA ~]# modprobe iptable_nat  
[root@hostA ~]# iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE  
[root@hostA ~]# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Este conjunto de comandos activará una NAT básica para la Internet. A fin de agregar soporte para FTP activo a través de esta puerta de acceso, ejecute lo que sigue:

```
[root@hostA ~]# modprobe ip_nat_ftp
```

Si está usando Fedora Core o Red Hat o RHEL y quiere hacer que la configuración de **iptables** sea parte de su script de arranque, ejecute lo siguiente:

```
[root@hostA ~]# /etc/rc.d/init.d/iptables save
```

NOTA Para los administradores de otras distribuciones de Linux, ustedes también pueden usar el comando **iptables-save** (el cual forma parte de la distribución de **iptables** y, por tanto, se aplica a todas las distribuciones de Linux). Este comando, en conjunción con **iptables-restore** les permitirá guardar y restablecer sus ajustes de **iptables**.

También necesitará hacer que el sistema realice en forma automática la línea “echo...” en la inicialización; agréguela a su script **/etc/rc.d/rc.local**.

Configuración de un firewall simple

En esta sección empezamos con un firewall de negación total para dos casos: una red simple en donde no están configurados servidores y la misma red, pero con algunos servidores configurados. En el primer caso, suponemos una red simple con dos lados: dentro de la red 10.1.1.0/24 (eth1) y la Internet (eth0). Advierta que por “servidor” queremos dar a entender cualquiera que necesite una conexión hecha *hacia* él. Esto podría significar un sistema Linux ejecutando ssh o un sistema Windows ejecutando BitTorrent.

Empecemos con el caso en donde no se tienen servidores que soportar.

En principio, necesitamos asegurarnos de que el módulo NAT está cargado y que el soporte FTP para NAT también está cargado. Hacemos eso con los comandos **modprobe**.

```
[root@hostA ~]# modprobe iptable_nat  
[root@hostA ~]# modprobe ip_nat_ftp
```

Con los módulos necesarios cargados, definimos las políticas predeterminadas para todas las cadenas. Para las cadenas INPUT, FORWARD y OUTPUT en la tabla filtro, fijamos el destino como DROP, DROP y ACCEPT, respectivamente. Para las cadenas POSTROUTING y PREROUTING, fijamos sus políticas predeterminadas en ACCEPT. Esto es necesario para que NAT funcione.

```
[root@hostA ~]# iptables -P INPUT DROP  
[root@hostA ~]# iptables -P FORWARD DROP  
[root@hostA ~]# iptables -P OUTPUT ACCEPT
```

```
[root@hostA ~]# iptables -t nat -P POSTROUTING ACCEPT
[root@hostA ~]# iptables -t nat -P PREROUTING ACCEPT
```

Con las políticas predeterminadas en su lugar, necesitamos definir la regla de línea de base del firewall. Lo que queremos realizar es sencillo: dejar que los usuarios que están dentro de la red (eth1) hagan conexiones con la Internet, pero no permitir que ésta haga conexiones de regreso. Para llevar a cabo esto, definimos una nueva cadena llamada "block" (bloquear) que usamos para agrupar nuestras reglas de seguimiento del rastro del estado. La primera regla en esa cadena sencillamente expresa que se permita pasar a cualquier paquete que sea parte de una conexión establecida o esté relacionada con una establecida. En la segunda se expresa que para que un paquete cree una nueva conexión no se puede originar desde la interfaz eth0 (que da la cara a la Internet). Si un paquete no se ajusta a cualquiera de estas dos reglas, la tercera regla fuerza a que se deje caer el paquete.

```
[root@hostA ~]# iptables -N block
[root@hostA ~]# iptables -A block -m state --state ESTABLISHED,RELATED -j ACCEPT
[root@hostA ~]# iptables -A block -m state --state NEW -i ! eth0 -j ACCEPT
[root@hostA ~]# iptables -A block -j DROP
```

Con la cadena de bloqueo en su lugar, necesitamos llamarla desde las cadenas INPUT y FORWARD. No nos preocupamos acerca de la cadena OUTPUT, ya que sólo los paquetes que se originan desde el propio firewall vienen de allí. Por otra parte, las cadenas INPUT y FORWARD necesitan ser verificadas. Recuerde que al realizar NAT, no se golpeará la cadena INPUT, de modo que necesitamos hacer que FORWARD realice la verificación. Si un paquete está destinado al propio firewall, necesitamos que las verificaciones se hagan desde la cadena INPUT.

```
[root@hostA ~]# iptables -A INPUT -j block
[root@hostA ~]# iptables -A FORWARD -j block
```

Por último, cuando el paquete sale del sistema, realizamos la función MASQUERADE desde la cadena POSTROUTING de la tabla NAT. Todos los paquetes que salen desde la interfaz eth0 pasan por esta cadena.

```
[root@hostA ~]# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Con todas las verificaciones de los paquetes y la manipulación detrás de nosotros, activamos el reenvío IP (algo indispensable para que NAT funcione) y la protección contra las cookies SYN, más activamos el comutador que evita que el firewall procese paquetes de difusión ICMP (ataques Smurf).

```
[root@hostA ~]# echo 1 > /proc/sys/net/ipv4/ip_forward
[root@hostA ~]# echo 1 > /proc/sys/net/ipv4/tcp_syncookies
[root@hostA ~]# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

En este punto tenemos un firewall trabajando para un entorno simple. Si no ejecuta servidores, puede guardar esta configuración y considerar que ha terminado. Por otra parte, supongamos que tiene dos aplicaciones que quiere hacer trabajar a través de este firewall: un sistema Linux en el interior de la red que necesita acceso ssh hacia y desde ubicaciones remotas y un sistema Windows desde el cual quiere ejecutar BitTorrent. Empecemos con el caso ssh.

Para hacer que un puerto quede disponible a través del firewall, necesitamos definir una regla que diga "si cualquier paquete en la interfaz eth0 (que da cara a la Internet) es TCP y tiene un puerto de destino de 22, cámbiese su dirección IP destino a que sea 172.16.1.3". Esto se realiza usando la acción DNAT en la cadena PREROUTING, ya que queremos cambiar la dirección IP del paquete antes que lo vea cualquiera de las otras cadenas.

El segundo problema que necesitamos resolver es cómo insertar una regla en la cadena FORWARD que permita que se autorice cualquier paquete cuya dirección IP destino sea 172.16.1.3 y su puerto destino sea el 22. La palabra clave es insertar (**-I**). Si anexamos la regla (**-A**) a la cadena FORWARD, el paquete por el contrario será dirigido a través de la cadena de bloqueo.

```
[root@hostA ~]# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 22 -j DNAT  
--to-destination 172.16.1.3  
[root@hostA ~]# iptables -I FORWARD -p tcp -d 172.16.1.3 --dport 22 -j ACCEPT
```

Podemos aplicar una idea semejante detrás de hacer que funcione BitTorrent. Supongamos que la máquina Windows que va a usar BitTorrent es la 172.16.1.2. El protocolo BitTorrent usa los puertos 6881-6889 para las conexiones que vienen de regreso al cliente. Por tanto, usamos un ajuste de rango de puertos en el comando **iptables**.

```
[root@hostA ~]# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6881:6889 -j DNAT --  
to-destination 172.16.1.2  
[root@hostA ~]# iptables -I FORWARD -p tcp -d 172.16.1.2 --dport 6881:6889 -j ACCEPT
```

¡Ta, ta! Ahora tiene un firewall en funcionamiento y soporte para un servidor ssh y un usuario de BitTorrent en el interior de su red. Debe sentirse tan feliz como para que ejecute la danza de la alegría.

RESUMEN

En este capítulo discutimos las entradas y salidas del firewall de Linux, Netfilter. Con esta información debe poder estructurar, mantener y administrar su propio firewall Linux.

Si no se ha vuelto evidente ya, Netfilter es un sistema imprevisiblemente complejo y rico. Los autores han escrito libros completos sólo sobre Netfilter y otros textos completos sobre los firewalls. En otras palabras, con este capítulo tiene un buen estuche de herramientas debajo de su cinturón, pero si en realidad desea sacar ventaja del alucinante poder de Netfilter, empiece a leer ahora; tiene una gran cantidad de páginas que recorrer. Además de este capítulo, puede ser que desee tomarse cierto tiempo para leer acerca de más detalles de Netfilter. Éstos se pueden hallar en <http://www.netfilter.org>. El libro *Firewalls and Internet Security: Repelling the Wily Hacker, Second Edition* de Cheswick, Bellovin y Rubin (Addison-Wesley, 2003) también es un buen texto.

No olvide que la seguridad también puede ser divertida. Tómese algo de tiempo para una lectura divertida con *The Cuckoo's Egg* por Clifford Stoll (Pocket, 2000). El libro es una historia verdadera de un astrónomo convertido en un cazador de hackers a finales de la década de 1980. Constituye una gran lectura y le da un sentido de lo que era la Internet antes de la comercialización, sin hablar de los firewalls.

CAPÍTULO 14



Seguridad local

Cuando usted escucha acerca de un nuevo ataque contra cualquier sistema operativo, una de sus primeras preguntas debe ser si es accesible o no a través de la red. Esto establece la distinción entre seguridad local y seguridad de la red, la cual, aunque relacionada, tiene dos enfoques diferentes para resolver el problema. En este capítulo nos enfocaremos sobre la seguridad local.

La *seguridad local* se encarga del problema de los ataques que requieren que el atacante sea capaz de hacer algo en el propio sistema con la finalidad de obtener acceso raíz. Por ejemplo, existe una clase completa de ataques que sacan ventaja de las aplicaciones que crean archivos temporales en el directorio **/tmp** pero no se comprueba la propiedad del archivo temporal, los permisos del archivo, o bien, si es un vínculo hacia otro archivo, antes de abrirlo o escribir en él. Un atacante puede crear un vínculo simbólico del nombre esperado del archivo temporal hacia un archivo que quiere corromper (por ejemplo, **/etc/passwd**) y ejecutar la aplicación; si ésta es SetUID para el raíz (lo que se cubre más adelante en este capítulo), destruirá el archivo **/etc/passwd** al escribir en su archivo temporal. El atacante puede usar la falta de **/etc/passwd** para sortear posiblemente otros mecanismos de seguridad de modo que pueda obtener acceso raíz.

Para un sistema que tiene en él usuarios indignos de confianza, éste puede ser un problema real. Los entornos universitarios a menudo son fruta madura para estos tipos de ataques, ya que los estudiantes necesitan tener acceso a los servidores para las tareas que se les asignan pero, al mismo tiempo, constituyen una gran amenaza para el sistema porque *a)* pueden aburrirse y *b)* no siempre piensan acerca de las consecuencias de sus acciones.

Los aspectos de seguridad local también se pueden disparar por aspectos de seguridad de la red. Si un aspecto de seguridad de la red da por resultado que un atacante pueda llamar cualquier programa o aplicación que esté en el servidor, puede realizar una hazaña basada en la seguridad local no sólo para darse pleno acceso al servidor sino para escalar sus propios privilegios hasta llegar al usuario raíz. Los “niñitos del script”, es decir, atacantes que usan los programas de ataque de otras personas porque son incapaces de crear el propio, utilizarán estas clases de métodos para obtener pleno acceso al sistema de usted. En su modo de hablar, “se apropiarán” de usted.

En este capítulo nos encargaremos de los fundamentos de mantener seguro su sistema contra los ataques hacia la seguridad local. No obstante, tenga presente que un solo capítulo sobre este tema no lo hará un experto. La seguridad es un campo que está en constante evolución y requiere actualización constante. La serie de libros Hacking Exposed es un lugar excelente a fin de que le pasen corriente para que arranque su conocimiento y la lista de correos BugTraq (<http://www.securityfocus.com/>) con frecuencia es donde se toman en primer lugar las grandes noticias sobre seguridad.

En el resto de este capítulo advertirás dos objetivos recurrentes: mitigar el riesgo y lo más sencillo es lo mejor. Lo primero es otra manera de ajustar su inversión (tanto en tiempo como en dinero, lo primero suele hacer que valga la pena lo último) dado el riesgo que está queriendo aceptar y el riesgo que plantea un servidor, si existe un término medio (un servidor Web que le sirve las fotografías para sus vacaciones en un vínculo de bajo ancho de banda es un riesgo más bajo que el de un servidor que maneja grandes transacciones financieras para Wall Street). El comentario “lo más sencillo es lo mejor” es ingeniería 101: los sistemas sencillos tienen menos propensión hacia los problemas, son más fáciles de arreglar, más fáciles de comprender y, de manera inevitable, son más confiables. Mantener sencillos sus servidores es un objetivo deseado.

FUENTES COMUNES DE RIESGO

Seguridad es la mitigación del riesgo. Con cada esfuerzo para mitigar el riesgo existe un costo asociado. Los costos no son de necesidad financieros; pueden tomar la forma de acceso restringido, pérdida de funcionalidad o tiempo. Su tarea como administrador es equilibrar los costos de mitigación del riesgo con el daño potencial que pueda causar un riesgo explotado.

Un ejemplo de equilibrio del riesgo se encuentra en el funcionamiento de un servidor Web. El riesgo de abrir un servicio que se puede sondear, al que se le pueden dirigir golpes y posiblemente explotar es inherente en la exposición de cualquier accesibilidad a la red. Empero, puede ser que encuentre que el riesgo de la exposición es bajo en tanto se dé mantenimiento al servidor Web y se parche de inmediato cuando surgen aspectos relativos a la seguridad. Si el beneficio de hacer funcionar un servidor Web es suficientemente grande como para justificar el costo para usted de mantenerlo, entonces es una empresa que vale la pena.

En esta sección lanzamos una mirada a las fuentes comunes de riesgo y examinamos qué puede hacer para mitigar esos riesgos.

Programas SetUID

Los programas SetUID son aquellos que se ejecutan con los permisos del propio programa, en lugar de con los permisos dados por el usuario que lo inicia. Esto permite a los administradores hacer que los usuarios normales dispongan de aplicaciones seleccionadas con privilegios más altos, sin tener que dar derechos administrativos a esos usuarios. Un ejemplo de un programa de ese tipo es **ping**. Debido a que la creación de paquetes en bruto en la red está restringida al usuario raíz (la creación de paquetes en bruto permite que la aplicación ponga cualquier contenido dentro de ese paquete, incluyendo ataques), la aplicación ping debe ejecutarse con el bit SetUID activado y el propietario fijado como el raíz. Por consiguiente, aun cuando el usuario "sshah" puede iniciar el programa, el programa ping se puede elevar hasta el usuario raíz para los fines de colocar un paquete ICMP sobre la red. La taquigrafía para este montaje es "SetUID root".

El problema con los programas que se ejecutan con los privilegios del raíz es que tienen una obligación de estar también muy conscientes de su seguridad. No debe ser posible para un usuario normal hacer algo peligroso en el sistema al usar ese programa. Esto quiere decir que es necesario escribir muchas verificaciones en el programa y tener que eliminar con todo cuidado los errores potenciales. Desde el punto de vista ideal, estos programas deben ser pequeños y hacer una cosa. Esto hace más fácil evaluar el código respecto a errores potenciales que pueden ser dañinos para el sistema o permitan al usuario obtener privilegios que no debe tener.

Desde una perspectiva cotidiana, está en el mejor interés del administrador mantener tan pocos programas raíz SetUID como sea posible. El equilibrio del riesgo aquí se da entre la disponibilidad de las características/funciones para los usuarios en comparación con el potencial de que sucedan cosas malas. Para algunos programas comunes, como **ping**, **mount**, **traceroute** y **su**, el riesgo es muy bajo en comparación con el valor que traen al sistema. Algunos programas SetUID bien conocidos, como el X Window System, plantean un riesgo de bajo a moderado; no obstante, dada la exposición que X Window ha tenido, es improbable que sea la raíz de cualesquier problemas. Si está haciendo funcionar un entorno de servidor puro, en donde no necesita X Window, nunca hace daño eliminarlo.

Los programas SetUID ejecutados por servidores Web casi siempre son una mala cosa. Tome grandes precauciones con estos tipos de aplicaciones y busque alternativas. La exposición es mucho mayor, ya que es posible que una entrada a la red (la cual puede venir de cualquier parte) dispare esta aplicación y afecte su ejecución.

Si encuentra que debe ejecutar una aplicación SetUID con privilegios del raíz, averigüe si es posible ejecutarla en un entorno chroot (que se discute más adelante en este capítulo).

Forma de hallar y crear programas SetUID

Un programa SetUID tiene un atributo especial de archivo que usa el núcleo para determinar si debe hacer caso omiso de los permisos predeterminados que se dieron a una aplicación. Al hacer una lista de un directorio, los permisos que se muestran en un archivo, en su salida `ls -l`, revelará este pequeño hecho. Por ejemplo,

```
[root@hostA ~]# ls -l /bin/ping
-rwsr-xr-x 1 root root 35108 Jun 15 2004 /bin/ping
```

Si la cuarta letra en el campo de permisos es una `s`, la aplicación es SetUID. Si el propietario del archivo es raíz, entonces la aplicación es raíz SetUID. En el caso de ping, podemos ver que se ejecutará con permisos del raíz disponibles para ella. Otro ejemplo es el programa Xorg (X Window):

```
[root@hostA ~]# ls -l /usr/X11R6/bin/Xorg
-rws--x--x 1 root root 1995032 Oct 20 14:45 /usr/X11R6/bin/Xorg
```

Como con ping, vemos que el cuarto carácter de los permisos es una `s` y el propietario es raíz. El programa Xorg es raíz SetUID.

Para determinar si un proceso en ejecución es SetUID, puede usar el comando `ps` con el fin de ver tanto el usuario real de un proceso y su usuario efectivo, de este modo:

```
[root@hostA ~]# ps ax -o pid,euser,ruser,comm
```

Esto dará como salida todos los programas en ejecución con su ID del proceso (PID), el usuario efectivo (euser), el usuario real (ruser) y el nombre del comando (comm). Si el usuario efectivo es diferente del real, es probable que sea un programa SetUID.

NOTA Algunas aplicaciones que son iniciadas por el usuario raíz *renuncian* a sus permisos para ejecutarse como un usuario inferior, con el fin de ayudar a la seguridad. Por ejemplo, el servidor Apache Web debe ser arrancado por el usuario raíz para escuchar el puerto TCP 80 (sólo los usuarios raíz pueden enlazarse a puertos con número menor a 1024), pero entonces renuncia a sus permisos del raíz e inicia todos sus enlaces como un usuario inferior [por lo común, el usuario "nobody" (nadie), "apache" o "www"].

Para hacer que un programa se ejecute como SetUID, use el comando `chmod`. Coloque el prefijo 4 a los permisos deseados para activar el bit SetUID (usando un prefijo de 2 se activará el bit SetGID, el cual es como el SetUID, pero con permisos de grupos, en lugar de permisos de usuarios). Por ejemplo, si tengo un programa llamado "myprogram" y quiero hacerlo raíz SetUID, hago lo siguiente:

```
[root@hostA ~]# chown root myprogram
[root@hostA ~]# chmod 4755 myprogram
```

```
[root@hostA ~]# ls -l myprogram  
-rws--x--x 1 root sshah 9812 Mar 12 14:29 myprogram
```

Hallar en dónde están esos programas SetUID puede ser engorroso esa primera vez por allí. Una instalación Fedora Core 3 completa puede tener con facilidad cientos de miles de archivos en el sistema (mi sistema tiene 365 685 archivos). Ir de directorio en directorio para hallar programas SetUID puede ser pesado y con propensión al error. De modo que, en lugar de hacer eso en forma manual, use el comando **find** de este modo:

```
[root@hostA ~]# find / -perm +4000 -ls
```

Procesos innecesarios

Al recorrer los scripts de arranque y paro, puede ser que haya advertido que un sistema Linux de edición estándar se inicia con una gran cantidad de procesos en ejecución. Lo que es necesario preguntarse es: *¿en realidad necesito todo lo que inicio?* El lector podría sorprenderse de lo que respondería.

Un ejemplo de la vida real: adelgazamiento del servidor del autor

Lancemos una mirada a un despliegue en la vida real de un servidor FreeBSD que maneja Web y correo electrónico hacia fuera de un firewall, y de un servidor Linux situado como una estación de trabajo de escritorio dentro del firewall, con un usuario de confianza. Las dos configuraciones representan extremos: configuración ajustada en un entorno hostil y una floja en un entorno bien protegido y de confianza (antes de que cualquiera mire con demasiada profundidad en el debate Linux contra FreeBSD, por favor advierta que la decisión de usar FreeBSD en el exterior se tomó con el resultado del lanzamiento de una moneda).

El sistema FreeBSD ejecuta la edición 4.9. Con los procesos innecesarios adelgazados, este sistema tiene 10 programas en ejecución con 18 procesos, cuando nadie se ha conectado. De los 10 programas, sólo SSH, Apache y Sendmail se ven desde el exterior en la red. El resto maneja funciones básicas de administración, como registro cronológico (syslog) y programación (cron). Con la eliminación de servicios no esenciales, usados sólo para experimentación (por ejemplo, servidor proxy Squid) y sólo disponibles mediante conexión al servidor, el conteo de programas cae hasta 7 (init, syslog, cron, SSH, Sendmail, Getty y Apache), con 13 procesos en ejecución, 5 de los cuales son Getty para soportar conexiones en los puertos en serie y el teclado. Como comparación, un sistema Fedora Core 3, configurado para uso en escritorio por parte de un usuario de confianza y que no ha sido adelgazado, tiene 40 procesos que manejan todo, desde el X Window System hasta la impresión para los servicios básicos de administración del sistema.

Para los sistemas de escritorio, en donde el riesgo está mitigado (por ejemplo, en donde el escritorio se asienta detrás de un firewall y los usuarios son de confianza), los beneficios de tener una gran cantidad de estas aplicaciones en ejecución bien valen la pena. Los usuarios de confianza aprecian contar con la capacidad de imprimir con facilidad y con el gozo de tener acceso a una buena interfaz del usuario. Para un servidor como el sistema FreeBSD 4.9, el riesgo sería demasiado grande para tener programas innecesarios en ejecución y, por consiguiente, no ha sido necesario eliminar todo.

El aspecto subyacente de la seguridad regresa al riesgo: ¿vale la pena el riesgo de ejecutar una aplicación por los beneficios que le trae? Si el valor que representa para usted lo que le brinda un proceso en particular es cero, porque no lo está usando, entonces no vale la cantidad de riesgo. Mirando más allá de la seguridad, existe el asunto práctico de la estabilidad y el consumo de recursos. Si un proceso representa para usted un valor de cero, incluso uno benigno que nada hace pero está ahí en un circuito ocioso, ocupa memoria, consume tiempo de procesador y recursos del núcleo. Si se encontrara un error en ese proceso, podría amenazar la estabilidad de su servidor. En suma: si no lo necesita, no lo ejecute.

Si su sistema se está ejecutando como servidor, minimice lo que ejecuta. Si no hay razón para que el servidor esté conectado a una impresora, desactive los servicios de impresión. Si no hay razón para que el servidor deba aceptar o enviar correos electrónicos, desactive el servidor de correo. Adelgácelo por completo, el servidor debe ejecutar lo mínimo preciso que necesita para proporcionar los servicios que se requieren de él. Por ejemplo, si no se ejecutan servicios a partir de **xinetd**, desactívelo. ¿No hay impresora? Desactive CUPS. ¿Ningún servidor de archivos? Desactive NFS y Samba. Un servidor Web que es responsable de proporcionar imágenes estáticas sólo necesita Apache, SSH, cron y syslog.

SUGERENCIA En la mayor parte de las distribuciones de Linux puede desactivar con rapidez los servicios con sólo renombrar el script de arranque en el directorio **rc.d**. Por ejemplo, si no necesita el servidor Apache y éste se inicia en el nivel de ejecución 3, sencillamente puede hacer lo siguiente:

```
[root@hostA /root]# cd /etc/rc.d/rc3.d  
[root@hostA rc3.d]# mv S15httpd K15httpd
```

Después de cambiar el primer carácter del script de arranque de *S* a *K*, ese script no se iniciará en el momento de la inicialización.

Elección del nivel correcto de ejecución con el cual inicializar

La mayor parte de las instalaciones predeterminadas de Linux se inicializarán directo hacia el X Window System. Esto da lugar a una bonita pantalla de arranque, un menú de conexión y una experiencia positiva total de escritorio. Sin embargo, para un servidor lo común es que todo eso sea innecesario, por las razones ya expresadas.

El nivel de ejecución que usa Linux para inicializar con X Window es el 5. Cambiar el nivel de ejecución hacia el 3 desactivará X Window y hacer que el sistema se inicialice en el nivel 3 evitirá que X Window arranque. Para hacer esto, edite el archivo **/etc/inittab** de modo que la línea

```
id:5:initdefault:
```

se cambie a

```
id:3:initdefault:
```

Advierta que el 5 cambió a 3.

SUGERENCIA Puede ver en qué nivel de ejecución se encuentra usted, basta que teclee **runlevel** en el mensaje. Por ejemplo,

```
[root@hostA /root]# runlevel  
N 3
```

Para forzar el cambio en el nivel de ejecución cuando el sistema se esté ejecutando, llame el comando **init** con el nivel deseado de ejecución como parámetro. Por ejemplo, para pasar hacia el nivel de ejecución 1 (modo de un solo usuario), ejecute

```
[root@hostA ~]# init 1
```

Programas que se ejecutan como raíz

Una vez que tiene lo esencial ejecutándose para hacer volar su servidor, es el momento de echar un vistazo para ver si cada programa se está ejecutando con sólo los permisos esenciales.

Mirando el aspecto del riesgo, queremos lograr el mayor beneficio de tener una aplicación exponiéndonos al mismo tiempo a la menor cantidad de riesgo. Por tanto, si una aplicación puede de entregar toda su potencialidad mientras se está ejecutando como un usuario que no es raíz, ejecútela como tal. Ya hemos visto que algunas aplicaciones, como Apache, hacen un esfuerzo por hacer esto. Otras aplicaciones bien conocidas del servidor que dejan caer los privilegios raíz, incluyen BIND (DNS) y SSH.

Por supuesto, no todo se puede ejecutar (o ejecutar con eficacia) sin los permisos del raíz. En estos casos es necesario intentar mitigar el riesgo. Hay tres cosas que puede usted hacer para ayudar a que suceda eso:

- ▼ Asegurarse de que está realizando la ejecución con un archivo detallado de configuración que se comporta exactamente como usted espera que lo haga.
- Mantenga actualizada la aplicación en todo momento. Si la aplicación tiene una lista de correos de “anuncio”, suscríbase a ella.
- ▲ Si es posible, ejecute la aplicación en un entorno chroot (vea más adelante en este capítulo).

El monitoreo de la página Web de una aplicación, listas de correos y otras fuentes de información, como BugTraq, puede ser tedioso, pero el tiempo y el riesgo que se evita al final vale la pena. Conforme crezca su sistema, puede ser que quiera considerar los sistemas de administración de parches, para ayudar en este esfuerzo.

Acceso otorgado a los usuarios

Los usuarios de un servidor no siempre corresponden a personas. Recuerde que todo proceso que se ejecute en un sistema Linux debe tener un propietario. La ejecución del comando **ps auxwww** en su sistema le mostrará todos los propietarios de los procesos en la columna de la extrema izquierda de su salida. Por ejemplo, en su sistema de escritorio, usted podría ser el único usuario humano, pero una mirada a los archivos **/etc/passwd** hace ver que se tienen 57 cuentas en el sistema.

Para que una aplicación deje caer sus privilegios del raíz, debe tener otro usuario que se puede ejecutar como tal. Aquí es donde esos usuarios adicionales entran en juego; cada aplicación que renuncia a ser del raíz, se le asigna otro usuario dedicado en el sistema. Este usuario se usa para que tenga la propiedad de todos los archivos de aplicación (incluyendo ejecutables, bibliotecas, de configuración y de datos), y se usa para que tenga la propiedad de los procesos de aplicación mientras se encuentran en ejecución. Al tener que cada aplicación que deje caer sus privilegios usa su propio usuario, se mitiga el riesgo de que una aplicación comprometida tenga acceso a otros archivos de configuración de aplicaciones. En esencia, se imponen limitaciones a

un atacante hacia qué archivos de aplicación tiene acceso, lo cual, dependiendo de la aplicación, puede prácticamente no ser interesante.

Recursos limitados

Para tener un mejor control de los recursos que se ponen a disposición de un usuario, puede usar los ajustes de los que se dispone en **ulimit** y que son configurables en forma global a través del archivo **/etc/security/limits.conf**. Con esto se puede restringir el número de archivos que pueden abrir, cuánta memoria pueden usar, el tiempo de CPU que pueden consumir y cuántos procesos pueden abrir. Los ajustes se leen por medio de las bibliotecas de PAM (Pluggable Authentication Module, Módulo enchufable de autenticación) cuando un usuario arranca.

La clave para estos ajustes es considerar la finalidad del servidor y la estación (si es aplicable a ésta). En el caso de un servidor, si una aplicación va a requerir que se ejecute una gran cantidad de procesos, asegúrese de tener un número suficiente de procesos disponibles para él. Para otros usuarios (como el servidor BIND DNS), nunca debe haber más de un pequeño manojo de procesos necesarios.

Antes de que se excite usted demasiado, aquí se requiere una pequeña advertencia: PAM debe tener posibilidad de ejecutarse para realizar los ajustes, antes que el usuario haga algo. Si la aplicación se inicia como raíz y, a continuación, deja caer los permisos, no es probable que se ejecute PAM. Desde un punto de vista práctico, esto significa que tener ajustes por separado por usuario no es probable que le dé a usted un resultado muy bueno en la mayor parte de los entornos de servidores. Lo que funcionará son ajustes globales que se apliquen tanto a los usuarios normales como al raíz. A fin de cuentas este detalle resulta ser algo bueno; tener al raíz bajo control ayuda a evitar que el sistema entre en espiral saliéndose de control, tanto por causa de ataques como por la de aplicaciones que fallen.

La fork bomb

Una broma común que todavía los estudiantes le juegan a otros estudiantes es entrar a las estaciones de trabajo de éstos y ejecutar una “fork bomb” (bomba de bifurcaciones). Éste es un programa que sencillamente crea tantos procesos que abruma el sistema y lo lleva a un rechinante alto. Para un estudiante, esto es molesto. Para un servidor de producción, esto es fatal. Una sencilla fork bomb basada en el shell, usando BASH, es

```
[sshah@hostA ~]$ while true; do sh -c sh & done
```

Si usted no tiene colocadas protecciones, este script hará que se caiga su servidor.

Lo interesante de las fork bombs es que no todas ellas son intencionales. Aplicaciones que fallan, sistemas con negación de ataques del servicio y, a veces, sólo sencillos errores tipográficos al introducir comandos puede hacer que sucedan cosas malas. Con el uso de los límites que se describen en este capítulo, puede mitigar el riesgo de una fork bomb al restringir el número máximo de procesos que puede llamar un solo usuario. Aun cuando la fork bomb todavía puede hacer que su sistema se cargue mucho, aún es probable que permanezca en condiciones de responder lo suficiente como para que usted ingrese y maneje la situación, esperando mantener todo ese tiempo los servicios ofrecidos. No es perfecto, pero resulta un equilibrio razonable entre tratar con los maliciosos y nada poder hacer.

El formato de cada línea en el archivo `/etc/security/limits.conf` es como sigue:

```
<domain>      <type>      <item>  <value>
```

Cualquier línea que empiece con un signo de número (#) es un comentario. El valor *domain* (dominio) contiene el nombre de acceso de un usuario o el nombre de un grupo. El *type* (tipo) se refiere al tipo de límite como "soft" (suave) o "hard" (estricto). El *item* (ítem) se refiere a lo que se aplica el límite. Se cuenta con los ítems siguientes, útiles para un administrador:

Ítem	Descripción	Predeterminado
fsize	Tamaño máximo del archivo	Ilimitado
nofile	Número máximo de archivos abiertos	1024
cpu	Cantidad máxima de tiempo (en minutos) que puede usar una CPU	Ilimitada
nproc	Número máximo de procesos que puede tener un usuario	Ilimitado
maxlogins	Número máximo de conexiones para un usuario	Ilimitado

Un ajuste razonable para la mayoría de los usuarios es sencillamente restringir el número de procesos, a menos que exista una razón específica para limitar los otros ajustes. Si necesita controlar el uso total del disco para un usuario, en lugar de ello debe usar cuotas del disco.

Un ejemplo para limitar el número de procesos a 128 para cada usuario sería

```
* hard nproc 128
```

Si usted se desconecta y se vuelve a conectar, puede ver el límite que tiene efecto al ejecutar el comando **ulimit** con el fin de ver cuáles son los límites

```
[root@hostA ~]# ulimit -a
core file size          (blocks, -c)  0
data seg size            (kbytes, -d)  unlimited
file size                (blocks, -f)  unlimited
pending signals          (-i)    1024
max locked memory        (kbytes, -l)  32
max memory size          (kbytes, -m)  unlimited
open files               (-n)    1024
pipe size                (512 bytes, -p) 8
POSIX message queues     (bytes, -q)   819200
stack size                (kbytes, -s)  10240
cpu time                 (seconds, -t)  unlimited
max user processes        (-u)    128
virtual memory            (kbytes, -v)  unlimited
file locks               (-x)  unlimited
```

El parámetro **-a** le dice a **ulimit** que haga una lista de todos los ajustes.

MITIGACIÓN DEL RIESGO

Una vez que sabe cuáles son los riesgos, mitigarlos se vuelve más fácil. Puede ser que encuentre que los riesgos que ve son suficientemente bajos como para que no se necesite tomar medidas adicionales de seguridad. Por ejemplo, un sistema de escritorio Windows XP usado por un usuario de confianza y con buena experiencia presenta un riesgo bajo como para usarse con privilegios de administrador. El riesgo de que el usuario descargue y ejecute algo que pueda causar daños al sistema es bajo. Además, los pasos tomados para mitigar el riesgo, como adherirse a sitios Web de mucha confianza y desactivar la descarga automática de archivos, alivian todavía más el riesgo. Este usuario bien experimentado puede hallar que poder ejecutar algunas herramientas adicionales y tener acceso sin restricciones al sistema bien valen la pena el riesgo de ejecutarse con privilegios de administrador. Como cualquier riesgo no trivial, la lista de advertencias es larga. Tan sólo redúzcase hasta una recompensa suficiente dada a un usuario que se ha tomado el tiempo para aprender cómo trabajar con seguridad.

Uso de chroot

La llamada `chroot()` del sistema permite a un proceso y a todos sus procesos hijos redefinir lo que perciben como el directorio raíz. Por ejemplo, si aplicara `chroot("/www")` e iniciara un shell, podría hallar que el uso del comando `cd` lo dejaría en `/www`. El programa lo tomaría como un directorio raíz, pero en realidad no lo sería. Esta restricción se aplica a todos los aspectos de comportamiento del proceso: en donde carga los archivos de configuración, las bibliotecas compartidas y los archivos de datos.

NOTA Una vez ejecutado, el cambio en el directorio raíz es irrevocable en toda la duración del proceso.

Al cambiar el directorio raíz percibido del sistema, un proceso tiene una visión restringida de lo que está en el sistema. No se encuentra disponible el acceso a otros directorios, bibliotecas y archivos de configuración. Debido a esta restricción, es necesario que una aplicación cuente con todos los archivos necesarios para funcionar, contenidos dentro del entorno chroot. Esto incluye cualesquiera archivos de contraseñas, bibliotecas, binarios y archivos de datos.

PRECAUCIÓN Un entorno chroot protegerá contra el acceso a archivos que están fuera del directorio, pero no protegerá contra la utilización del sistema, el acceso a la memoria, el acceso al núcleo y la comunicación entre procesos. Esto quiere decir que se tiene vulnerabilidad en la seguridad de la que pueden sacar ventaja al enviar señales a otro proceso, que será posible explotarlo desde el interior de un entorno chroot. En otras palabras, chroot no es una cura perfecta sino, más bien, un excelente elemento disuasorio.

Cada aplicación necesita su propio conjunto de archivos y ejecutables y, por consiguiente, las direcciones para hacer que una aplicación funcione en un entorno chroot varían. No obstante, el principio sigue siendo el mismo: hacerlo del todo autocontenido bajo un solo directorio con una estructura postiza de directorio raíz.

Un ejemplo de entorno chroot

Como ejemplo, creamos un entorno chroot para el shell BASH. Empezamos por crear el directorio en el que queremos poner todo. Como éste sólo es un ejemplo, crearemos un directorio en /tmp llamado **myroot**.

```
[root@hostA ~]# mkdir /tmp/myroot  
[root@hostA ~]# cd /tmp/myroot
```

Supongamos que sólo necesitamos dos programas: **bash** y **ls**. Creamos el directorio **bin** bajo **myroot** y copiamos los binarios allí.

```
[root@hostA myroot]# mkdir bin  
[root@hostA myroot]# cp /bin/bash bin  
[root@hostA myroot]# cp /bin/ls bin
```

Con los binarios allí, necesitamos comprobar si éstos necesitan algunas bibliotecas. Usemos el comando **ldd** para determinar cuáles bibliotecas (si las hay) usan estos dos programas.

```
[root@hostA myroot]# ldd /bin/bash  
libtermcap.so.2 => /lib/libtermcap.so.2 (0x00b9d000)  
libdl.so.2 => /lib/libdl.so.2 (0x0098e000)  
libc.so.6 => /lib/tls/libc.so.6 (0x00840000)  
/lib/ld-linux.so.2 (0x00823000)  
[root@hostA myroot]# ldd /bin/ls  
librt.so.1 => /lib/tls/librt.so.1 (0x006a8000)  
libacl.so.1 => /lib/libacl.so.1 (0x00abd000)  
libselinux.so.1 => /lib/libselinux.so.1 (0x0062b000)  
libc.so.6 => /lib/tls/libc.so.6 (0x00840000)  
libpthread.so.0 => /lib/tls/libpthread.so.0 (0x00a7f000)  
/lib/ld-linux.so.2 (0x00823000)  
libattr.so.1 => /lib/libattr.so.1 (0x00ab7000)
```

Ahora que sabemos cuáles bibliotecas necesitamos que estén en su lugar, creamos los directorios **lib** y **lib/tls** y copiamos las bibliotecas en ellos.

```
[root@hostA myroot]# mkdir lib  
[root@hostA myroot]# mkdir lib/tls  
[root@hostA myroot]# cp /bin/bash bin  
[root@hostA myroot]# cp /lib/libtermcap.so.2 lib  
[root@hostA myroot]# cp /lib/libdl.so.2 lib  
[root@hostA myroot]# cp /lib/libc.so.6 lib  
[root@hostA myroot]# cp /lib/ld-linux.so.2 lib  
[root@hostA myroot]# cp /lib/tls/librt.so.1 lib/tls
```

```
[root@hostA myroot]# cp /lib/libacl.so.1 lib
[root@hostA myroot]# cp /lib/libselinux.so.1 lib
[root@hostA myroot]# cp /lib/tls/libc.so.6 lib/tls
[root@hostA myroot]# cp /lib/tls/libpthread.so.0 lib/tls
[root@hostA myroot]# cp /lib/libattr.so.1 lib
```

Fedora Core incluye un pequeño programa llamado **chroot** que pide la llamada del sistema por nosotros, de modo que no necesitamos escribir nuestro propio programa C para hacerlo. Admite dos parámetros: el directorio que queremos hacer que sea el directorio raíz y el comando que queremos ejecutar en el entorno chroot. Queremos usar **/tmp/myroot** como el directorio y arrancar **/bin/bash**, por tanto:

```
[root@hostA myroot]# chroot /tmp/myroot /bin/bash
```

Debido a que no existen **/etc/profile** ni **/etc/bashrc** para cambiar nuestro mensaje, ésta cambiará a **bash-3.00#**. Ahora intentemos un **ls**:

```
bash-3.00# ls
bin  lib
```

Enseguida intentemos un **pwd** para ver el directorio actual de trabajo:

```
bash-3.00# pwd
/
```

Como no contamos con un archivo **/etc/passwd** ni con un **/etc/group**, un comando **ls -l** mostrará los valores UID para cada archivo. Por ejemplo,

```
bash-3.00# cd lib
bash-3.00# ls -l
-rwxr-xr-x 1 0 0 108332 Mar 20 23:52 ld-linux.so.2
-rwxr-xr-x 1 0 0 23572 Mar 20 23:57 libacl.so.1
-rwxr-xr-x 1 0 0 26319 Mar 20 23:58 libattr.so.1
-rwxr-xr-x 1 0 0 1504728 Mar 20 23:51 libc.so.6
-rwxr-xr-x 1 0 0 16908 Mar 20 23:51 libdl.so.2
-rwxr-xr-x 1 0 0 56288 Mar 20 23:57 libselinux.so.1
-rwxr-xr-x 1 0 0 12592 Mar 20 23:51 libtermcap.so.2
drwxr-xr-x 2 0 0 4096 Mar 20 23:57 tls
```

Con nada más que usar, el entorno no es terriblemente útil como para el trabajo práctico, lo que lo hace magnífico desde la perspectiva de la seguridad; sólo damos los archivos mínimos necesarios para que una aplicación funcione, minimizando de esta manera nuestra exposición en el caso de que la aplicación se ponga en peligro. Tenga presente que no todos los entornos chroot necesitan tener un shell y un comando **ls** instalados; si el servidor BIND DNS sólo necesita instalados sus propios ejecutables, bibliotecas y archivos de zona, entonces eso es todo lo que usted necesita.

SELinux

La National Security Agency (NSA, Agencia Nacional de Seguridad) del gobierno de Estados Unidos ha tomado un papel público cada vez mayor en la seguridad de la información, en especial debido a la preocupación creciente acerca de los ataques a la seguridad de la información que podrían plantear una seria amenaza a la capacidad mundial para funcionar. ¿Puede alguno de nosotros imaginar el caos si los mercados financieros del mundo fueran atacados con éxito?

Con Linux convirtiéndose en un componente clave cada vez más importante de la computación empresarial, la NSA planteó la creación de un conjunto de parches para incrementar la seguridad de Linux. Los parches se han emitido bajo licencia GPL con código fuente completo y, como consecuencia, sujetos al escrutinio del mundo, un aspecto importante dadas la presencia a escala mundial de Linux y la comunidad desarrolladora. Los parches se conocen en forma colectiva como "SELinux", acrónimo de "Security-Enhanced Linux" (Linux con seguridad mejorada).

Muchas distribuciones de Linux, incluyendo Fedora Core y Red Hat, han aceptado estos parches como parte de su distribución normal del núcleo. Esto ha hecho que los parches y las mejoras sean de largo alcance, así como un beneficio global para la comunidad Linux.

El concepto subyacente que se encuentra detrás de los parches es que todos los objetos dentro del sistema, tanto el núcleo como el espacio de los usuarios, tienen un Security Identifier (SID, Identificador de seguridad) asociado con ellos. Entonces, estos SID se asocian con las políticas que rigen cuáles acciones se pueden tomar sobre el objeto y cuáles no, por parte de un usuario dado. Debido a lo extremo granuloso de estos objetos, es posible expresar reglas muy ricas y complejas que gobiernen el modelo de seguridad y el comportamiento de un sistema Linux.

El alcance completo de los parches se encuentra más allá del alcance de una sola sección de este libro. Si el lector tiene interés en estos parches, visite el sitio Web de SELinux en <http://www.nsa.gov/selinux>.

MONITOREO DE SU SISTEMA

A medida que usted se familiarice con Linux, sus servidores y su operación cotidiana, encontrará que empieza a adquirir cierta "sensación" de lo que es normal. Esto puede sonar peculiar pero, de modo muy semejante a la manera en que aprende a "sentir" cuando su automóvil no está bastante bien, sabrá cuándo su servidor tampoco lo está.

Parte de la obtención de una sensación del sistema requiere un monitoreo básico del mismo. Para el comportamiento del sistema local, esto requiere que usted confíe en su sistema subyacente como si no hubiera sido puesto en peligro de manera alguna. Si en realidad su servidor se ve en peligro y se instala un "juego raíz de instrumentos" que le haga un rodeo a los sistemas de monitoreo, puede ser muy difícil ver lo que está sucediendo. Por esta razón, una mezcla de monitoreo basado en el anfitrión y en un anfitrión remoto es una buena idea.

Registro cronológico

De modo predeterminado, la mayor parte de sus archivos de registro se almacenarán en el directorio `/var/log`, con logrotate haciendo girar de manera automática las entradas de registro de modo regular. Aun cuando resulta práctico realizar el registro en su disco local, con frecuencia es una mejor idea hacer que su sistema envíe sus entradas de registro hacia un servidor dedicado para ello, que sólo maneje datos syslog, y posiblemente a un servidor SSH. Teniendo activado

el registro cronológico remoto, puede tener la certeza de que cualesquiera entradas de registro enviadas al servidor para el efecto, antes de un ataque, están 100% garantizadas de no ser manipuladas indebidamente.

Debido al volumen de datos de registro que es posible generar, puede hallar que sea prudente aprender algunas habilidades básicas de creación de scripts, de modo que pueda analizar con facilidad esos datos y hacer resaltar/enviar por correo todo lo que sea peculiar o merecedor de una sospecha. Por ejemplo, un filtro que genere correos relativos a los registros resulta útil sólo para un administrador. Esto permite a este último seguirle la huella tanto a la actividad normal como a la errónea, sin tener que leer todos los días de un lado a otro un número significativo de mensajes de registro.

Uso de ps y netstat

Una vez que tenga su servidor montado y funcionando, tómese un momento para estudiar la salida del comando **ps auxww**. Las desviaciones de su salida deben captar su atención en el futuro. Como parte del monitoreo, puede hallar útil hacer una lista periódica de cuáles procesos se encuentran en ejecución y tener conocimiento de cualesquier procesos que usted no espera que estén allí por alguna razón. Sospeche en especial de cualesquier programas de captura de datos, como **tcpdump**, que no arrancó usted.

Lo mismo se puede decir acerca de la salida del comando **netstat -an**. Una vez que tenga cierto sentido de lo que representa el tráfico normal y los puertos abiertos de manera normal, cualesquier desviaciones respecto de esa salida deben disparar el interés hacia por qué está allí la desviación. ¿Alguien cambió la configuración del servidor? ¿La aplicación hizo algo inesperado? ¿Existe actividad amenazante en el servidor?

Entre **ps** y **netstat**, debe tener un manejo justo sobre los sucesos relativos a su red y su lista de procesos.

Uso de df

El comando **df** muestra el espacio disponible en cada una de las particiones del disco que están montadas. La ejecución de **df** de manera regular, con el fin de ver la rapidez con la que se usa el espacio de disco, es una buena manera de ver si hay alguna actividad cuestionable. Un cambio repentino en la utilización del disco debe hacer saltar la curiosidad hacia el lugar del que provino el cambio. En el pasado hemos encontrado que algunos usuarios han empezado a usar sus directorios de inicio para almacenar vastas cantidades de archivos MP3. Dejando a un lado los aspectos legales, no estuvimos demasiado felices acerca de las repercusiones que tuvo sobre nuestros respaldos, y si éstos fallaran porque la cinta funcionaba con espacio tomado para almacenar los archivos de música de alguien, en lugar de los archivos clave necesarios para la empresa, es evidente que tendríamos un desastre en nuestras manos. En el sentido de la seguridad, si los tamaños de los directorios Web o FTP crecen de modo significativo sin razón, puede haber un asomo de problemas con el uso no autorizado de su servidor.

Listas de correos

Como parte de la administración de la seguridad de su sistema, debe suscribirse a listas clave de correo relativo a la seguridad, como BugTraq (<http://www.securityfocus.com/archive/1>). BugTraq es una lista moderada de correos que genera sólo un pequeño manojo de correos al día, la mayor parte de los cuales no pertenecerán al software que usted está ejecutando. Sin embargo,

Obtención de una sensación de mi servidor

Los administradores tienen incontables relatos de cómo supieron que algo no estaba suficientemente correcto. Una de esas historias es la de un administrador que administra un sistema Sun IPX desde hace muchos años. Cuando iba caminando hacia la sala del servidor, el administrador supo que algo no andaba bien porque la actividad del disco era suficientemente frenética como para *escucharse*, cuando debía haber sido silenciosa. Al ir de inmediato a la consola correspondiente, el tiempo de respuesta al hacer entrar la petición de acceso al sistema y la obtención de un mensaje del shell también estaba fuera de lo normal; era evidente que el servidor estaba bajo algún tipo de carga significativa. Con el disco trabajando con esa intensidad, el administrador ejecutó el programa `df` para ver cuánto espacio libre había. Con seguridad suficiente, el carrete de correo estaba cerca de quedar sin espacio, lo cual estaba de manera significativa fuera de lo que había sido el día anterior. Menos de un minuto después se encontró el culpable: alguien había enviado una distribución masiva de correo que hizo que se copiara cientos de veces un archivo de 40MB. El remitente tuvo que absorber 1GB de espacio (lo cual fue enorme en el momento) antes que se purgara el correo y el sistema regresara a lo normal. Con el sistema comportándose de nuevo, todo lo que constituyó la *sensación* del sistema, desde los sonidos producidos hasta el tiempo de respuesta de los mensajes, había regresado a lo normal.

ésta es donde es probable que se muestren primero los sucesos críticos. Los varios últimos gusanos significativos que atacaron los anfitriones de Internet se dieron a conocer en tiempo real en estas listas de correos.

Además de BugTraq, cualesquiera listas de seguridad para el software del que usted está a cargo, son su responsabilidad. Busque también listas de anuncios para el software que usa. Todas las distribuciones importantes de Linux también mantienen listas de anuncios para los aspectos de seguridad que pertenecen a sus distribuciones específicas. Los principales vendedores de software también mantienen sus propias listas. Por ejemplo, Oracle mantiene su información en línea a través de su portal Web MetalLink y las listas de correos correspondientes.

Aun cuando esto puede parecer una gran cantidad de correo, considere que la mayor parte de las listas que se basan en anuncios son en extremo de bajo volumen. En general, usted no debe encontrarse en la necesidad de tratar con una cantidad significativamente mayor de correo que aquella con la que ya lo hace.

RESUMEN

En este capítulo aprendió acerca de la forma de hacer seguro su sistema Linux, la mitigación del riesgo y la manera de aprender qué buscar al tomar decisiones respecto a cómo equilibrar las características/funciones con la necesidad de contar con seguridad. En específico, cubrimos las causas del riesgo, como los programas SetUID, los que se ejecutan como raíz y los innecesarios. También cubrimos los procedimientos para mitigar el riesgo a través del uso de entornos chroot

y el control del acceso para los usuarios. Por último discutimos algunas de las cosas que deben monitorearse como parte de la limpieza diaria de la casa.

Al final encontrará que mantener un entorno razonablemente seguro es, en gran parte, algo de buena higiene. Conserve limpio su servidor de aplicaciones innecesarias, asegúrese de minimizar el entorno para cada aplicación de manera que se limite la exposición y parche su software conforme salgan a la luz aspectos relativos a la seguridad. Con estas tareas básicas encontrará que sus servidores son bastante confiables y seguros.

En una nota final tenga presente que esta sección por sí sola no lo convierte en un experto en seguridad, en gran parte como el capítulo sobre los firewalls de Linux no lo convirtieron en un experto en éstos. Linux siempre está evolucionando y mejorando. Necesitará continuar haciendo un esfuerzo para aprender acerca de lo más reciente y ampliar su conocimiento sobre la seguridad en general.

CAPÍTULO 15



Seguridad en la red

En el capítulo 14 se hizo la afirmación: "Cuando escuche acerca de un nuevo ataque contra cualquier sistema operativo, una de sus primeras preguntas debe ser si es accesible o no a través de la red". La respuesta a la pregunta tuvo un apoyo en la manera en que se enfoca el ataque. En otras palabras, ¿el ataque requiere acceso local al sistema o el ataque sólo necesita conectividad a la red que puede enviar paquetes hacia el sistema? El primer caso se cubrió en el capítulo 14; el último se cubre en este capítulo.

La *seguridad en la red* se encarga del problema de los atacantes que envían tráfico malicioso por la red hacia su sistema, con el intento de hacer que no se disponga de su sistema (ataque de negación del servicio) o explotar las debilidades de su sistema para obtener acceso al mismo ("echar raíces" en su sistema). Empero, la seguridad en la red no es un reemplazo de los procedimientos de seguridad local que se discutieron en el capítulo anterior. Tanto los procedimientos de seguridad local como aquellos para la red son necesarios para mantener las cosas funcionando de la manera en que usted lo espera.

En este capítulo cubrimos cuatro aspectos de la seguridad en la red: seguimiento del rastro a los servicios, monitoreo de los servicios de la red, manejo de los ataques y herramientas para realizar pruebas. Estas secciones deben utilizarse en conjunción con el capítulo anterior referente a la seguridad local, así como con el capítulo 13.

TCP/IP Y SEGURIDAD EN LA RED

En este capítulo se supone que usted tiene experiencia en la configuración de un sistema para usarse en una red TCP/IP. En virtud de que el enfoque de lo que sigue es sobre la seguridad en la red y no una introducción a la operación en red, en esta sección sólo se discuten aquellas partes de TCP/IP que afectan la seguridad de su sistema. Si usted tiene curiosidad acerca de los trabajos internos de TCP/IP, lea el capítulo 11.

Importancia de los números de los puertos

Cada anfitrión en una red basada en IP tiene por lo menos una dirección IP. Además, cada anfitrión basado en Linux tiene muchos procesos en ejecución por separado. Cada proceso tiene el potencial de ser un cliente de la red, un servidor de ésta o las dos cosas. Con la potencialidad de que más de un proceso es capaz de actuar como servidor en un solo sistema, no es suficiente el uso de una sola dirección IP para identificar una conexión a la red.

Con el fin de resolver este problema, TCP/IP agrega un componente que identifica un *puerto* TCP (o UDP). Cada conexión de un anfitrión a otro tiene un *puerto fuente* y un *puerto destino*. Cada puerto se nombra con un entero entre 0 y 65535.

Para identificar cada conexión única posible entre dos anfitriones, el sistema operativo le sigue el rastro a cuatro piezas de información: la dirección IP fuente, la dirección IP destino, el número del puerto fuente y el número del puerto destino. Está garantizado que la combinación de estos cuatro valores es única para todas las conexiones anfitrión a anfitrión (en realidad, el sistema operativo le sigue el rastro a una miríada de información de conexiones, pero sólo se necesitan estos cuatro elementos para identificar de manera única una conexión).

El anfitrión que inicia una conexión especifica la dirección IP y el número del puerto destino. Es obvio que ya se conoce la dirección IP fuente. Pero el número del puerto fuente, el valor que hará que la conexión sea única, la asigna el sistema operativo fuente. Busca en toda su lista de

conexiones ya abiertas y asigna el siguiente número disponible de puerto. Por convención, este número siempre es mayor que 1024 (los números de puerto del 0 al 1023 se reservan para usos del sistema). Técnicamente, el anfitrión puente también puede seleccionar su número de puerto fuente. Sin embargo, para hacer esto, otro proceso no puede haber tomado ya ese puerto. En general, la mayor parte de las aplicaciones dejan que el sistema operativo seleccione por ellas el número del puerto fuente.

Conociendo esta disposición, podemos ver de qué manera el anfitrión A puede abrir múltiples conexiones para un solo servicio en el anfitrión B destino. La dirección IP y el número de puerto del anfitrión B siempre serán constantes, pero el número de puerto del anfitrión A será diferente para cada conexión. Por lo tanto, la combinación de los IP y números de puertos, fuente y destino (un conjunto de cuatro números) es única, y los dos sistemas pueden tener múltiples corrientes (conexiones) independientes de datos entre sí.

Para que un servidor ofrezca servicios, debe ejecutar programas que escuchen los números específicos de puertos. Muchos de estos números de puertos se llaman *servicios bien conocidos*, porque el número de puerto asociado con un servicio es un estándar aprobado. Por ejemplo, el puerto 80 es el puerto de servicio bien conocido para el protocolo HTTP.

En “Uso del comando netstat” miraremos el comando **netstat** como una herramienta importante para la seguridad en la red. Cuando tenga una comprensión firme de qué representan los números de puerto, será capaz de identificar e interpretar con facilidad las estadísticas de seguridad en la red proporcionadas por el comando **netstat**.

SEGUIMIENTO DEL RASTRO DE LOS SERVICIOS

Los servicios proporcionados por un servidor son los que lo hacen servidor. Estos servicios los realizan procesos que se enlazan a los puertos de la red y escuchan las peticiones que entran. Por ejemplo, un servidor Web podría iniciar un proceso que se enlaza al puerto 80 y escucha las peticiones para descargar las páginas de un sitio. A menos que exista un proceso que escuche a un puerto específico, Linux sencillamente ignorará los paquetes enviados a ese puerto.

En esta sección se discute el uso del comando **netstat**, una herramienta para seguir el rastro de las conexiones a la red (entre otras cosas) en su sistema. Sin duda, es una de las herramientas más útiles para la corrección de errores en su arsenal para la detección de fallas de seguridad y los problemas cotidianos de la red.

Uso del comando netstat

Para seguir el rastro de cuáles puertos están abiertos y cuáles tienen procesos que los están escuchando, usamos el comando **netstat**. Por ejemplo,

```
[root@hostA ~]# netstat -natu
```

Active Internet connections (servers and established)					
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	0.0.0.0:32768	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:111	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:113	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN

tcp	0	0	127.0.0.1:5335	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:25	0.0.0.0:*	LISTEN
tcp	0	0	:::22	:::*	LISTEN
tcp	0	132	192.168.1.4:22	192.168.1.33:2129	ESTABLISHED
udp	0	0	0.0.0.0:32768	0.0.0.0:*	
udp	0	0	0.0.0.0:813	0.0.0.0:*	
udp	0	0	0.0.0.0:5353	0.0.0.0:*	
udp	0	0	0.0.0.0:5353	0.0.0.0:*	
udp	0	0	0.0.0.0:111	0.0.0.0:*	
udp	0	0	0.0.0.0:631	0.0.0.0:*	

De modo predeterminado (sin parámetros), **netstat** proporcionará todas las conexiones establecidas, tanto para la red como para enchufes de dominios. Eso quiere decir que no sólo veremos las conexiones que en realidad están trabajando sobre la red, sino también las comunicaciones entre procesos (las cuales, desde el punto de vista del monitoreo de la seguridad, no son útiles). De modo que en el comando que se acaba de ilustrar, le hemos pedido a **netstat** que nos muestre todos los puertos (**-a**), sea que estén escuchando o en realidad estén conectados, para TCP (**-t**) y UDP (**-u**). Le hemos dicho a **netstat** que no consuma tiempo resolviendo las direcciones IP en nombres de anfitriones (**-n**).

En la salida de **netstat**, cada línea representa un puerto TCP o UDP de la red, como se indica en la primera columna de la salida. En la columna Recv-Q (recieve queue, lista de recepción) se tiene una lista del número de bytes recibidos por el núcleo, pero no leídos por el proceso. Enseguida, la columna Send-Q nos dice el número de bytes enviados hacia el otro lado de la conexión, pero que no han sido reconocidos.

Las columnas cuarta, quinta y sexta son las más interesantes en términos de la seguridad del sistema. La columna Local Address (Dirección local) le dice la dirección IP y el número de puerto de su propio servidor. Recuerde que su servidor se reconoce como 127.0.0.1 y 0.0.0.0, así como también su dirección IP. En el caso de múltiples interfaces, cada puerto al que se está escuchando se mostrará en las dos interfaces y, por consiguiente, como dos direcciones IP separadas. El número del puerto está separado de la dirección IP por medio de dos puntos. En la salida del ejemplo de **netstat** que acaba de mostrarse, el dispositivo Ethernet tiene la dirección IP 192.168.1.4.

La quinta columna, Foreign Address (Dirección extranjera) identifica el otro lado de la conexión. En el caso de un puerto al que se le está escuchando por nuevas conexiones, el valor predeterminado será 0.0.0.0:*. Esta dirección IP no tiene significado, ¡ya que todavía estamos esperando que un anfitrión remoto se conecte con nosotros!

La sexta columna nos dice el State (Estado) de la conexión. En la página man de **netstat** se encuentra una lista de todos los estados, pero los dos que verá con la mayor frecuencia son LISTEN (escuchar) y ESTABLISHED (establecida). El estado LISTEN significa que hay un proceso en su servidor escuchando al puerto y listo para aceptar nuevas conexiones. ESTABLISHED sólo significa que se encuentra establecida una conexión entre el cliente y el servidor.

Implicaciones de seguridad de la salida de netstat

Al hacer una lista de todas las conexiones de las que se dispone, puede obtener una instantánea de lo que está haciendo el sistema. Debe ser capaz de explicar y justificar *todos* los puertos que se encuentran en la lista. Si su sistema está escuchando a un puerto que usted no puede explicar, esto debe provocar sospechas.

Si ha estado usando sus celdas de memoria para otras finalidades y no ha memorizado los servicios y los números de puertos asociados, puede mirar la información correlativa que necesita en el archivo `/etc/services`. Sin embargo, algunos servicios (los más notables son aquellos que usan el programa portmapper) no tienen fijados números de puertos pero son servicios válidos. Para ver cuál proceso está asociado con un puerto, use la opción `-p` con `netstat`. Esté a la caza de procesos extraños o desacostumbrados que usen la red. Por ejemplo, si el shell BASH está escurriendo un puerto de la red, puede tener una amplia certeza de que algo raro está pasando.

Por último, recuerde que usted sólo está interesado en el puerto destino de una conexión; éste le dice a cuál servicio se está conectando y si es legítimo. Por desgracia, `netstat` no nos dice en forma explícita quién originó una conexión, pero por lo común podemos figurárnoslo si reflexionamos un poco. Por supuesto, familiarizarse con las aplicaciones que en realidad usted ejecuta y su uso de los puertos de la red es la mejor manera de determinar quién originó una conexión hacia dónde. En general, encontrará que la regla empírica es que el lado cuyo número de puerto es mayor que 1024 es aquel en el que se originó la conexión. Es obvio que esta regla general no se aplica a los servicios que es típico que se ejecuten en puertos de número mayor que 1024, como X Window (puerto 6000).

Liga a una interfaz

Un procedimiento común para mejorar la seguridad de un servicio que se ejecuta en un servidor es hacerlo de modo que sólo se ligue a una interfaz específica de la red. De modo predeterminado, las aplicaciones se ligarán a todas las interfaces (visto como 0.0.0.0 en la salida de `netstat`). Esto permitirá una conexión a ese servicio desde cualquier interfaz en tanto que la conexión pase por cualesquier firewall Netfilter que pueda usted haber configurado. Sin embargo, si sólo necesita que se disponga de un servicio en una red particular, debe configurar ese servicio para que se ligue a una interfaz específica.

Por ejemplo, supongamos que existen tres interfaces en su servidor: eth0, la cual es la 192.168.1.4; eth1, la cual es la 172.16.1.1 y lo, la cual es la 127.0.0.1. Supongamos también que su servidor no tiene activado el reenvío IP (`/proc/sys/net/ipv4/ip_forward`). En otras palabras, las máquinas en el lado de 192.168.1.0/24 no se pueden comunicar con las máquinas en el lado de 172.16/16. La red 172.16/16 (eth1) representa la red “segura” y, por supuesto, 127.0.0.1 representa el propio anfitrión.

Si la aplicación se liga a 172.16.1.1, entonces sólo aquellas aplicaciones en el lado de 172.16/16 podrán alcanzar la aplicación y conectarse a ella. Si no confía en los anfitriones en el lado de 192.168.1/24 (por ejemplo, es un DMZ), entonces ésta es una manera segura para dar servicio a un segmento sin que se exponga usted mismo al otro. Para tener incluso una menor exposición, puede ligar una aplicación a 127.0.0.1. Al hacerlo, dispone que las conexiones tengan que originarse desde el propio servidor para comunicarse con el servicio. Por ejemplo, si necesita ejecutar la base de datos MySQL para una aplicación basada en la red, y la aplicación se ejecuta en el servidor, entonces al configurar MySQL para que sólo acepte conexiones desde 127.0.0.1 significa que se mitiga de manera significativa cualquier riesgo asociado con conectarse en forma remota a ella y explotar el servicio MySQL. El atacante tendría que comprometer su aplicación basada en la Web y, de alguna manera, hacer consultar la base de datos en su beneficio (ataque de inyección SQL).

SUGERENCIA Si necesita suministrar un servicio a un grupo de usuarios técnicamente competentes, a través de la Internet, ligar un servicio a un anfitrión local y, entonces, forzar al grupo a que use túneles SSH es una gran manera de requerir acceso autenticado y codificado hacia el servicio. Por ejemplo, a mi esposa le gusta descargar su correo a través de POP. Debido a que quiere tener acceso a su correo desde un lugar remoto, finalizamos ligando el servidor POP al anfitrión local y, entonces, ella usó SSH con reenvío del puerto para conectarse desde su Macintosh para tener acceso al servicio (`ssh -l username -L 1110:127.0.0.1:1110 servername`). Luego ella fijó su lector de correo POP para bajarlo desde 127.0.0.1:1110 a su propia máquina. Cualquiera que ejecute un escaneador de puertos contra mi servidor no verá el servidor de correo POP.

Interrupción de servicios

Una finalidad del comando `netstat` es determinar cuáles servicios están activados en los servidores de usted. Hacer que Linux sea más fácil de instalar y administrar precisamente fuera de la caja ha conducido a más y más ajustes predeterminados que son inseguros, de modo que es importante de manera especial seguir el rastro de los servicios.

Cuando esté evaluando cuáles servicios deben permanecer y cuáles deben desecharse, responda las siguientes preguntas:

1. *¿Necesitamos el servicio?* La respuesta a esta pregunta es muy importante. En la mayor parte de las situaciones debe poder desactivar un gran número de servicios que arrancan de modo predeterminado. Por ejemplo, un servidor Web que sólo es eso no debe necesitar ejecutar NFS.
2. *Si no necesitamos el servicio, ¿es seguro el ajuste predeterminado?* Esta pregunta también puede ayudarle a eliminar algunos servicios; si no son seguros y no se pueden hacer seguros, entonces hay posibilidades de que deban eliminarse. Por ejemplo, si el acceso remoto es un requisito y Telnet es el servicio activado para suministrar esa función, entonces en su lugar debe usarse una alternativa como SSH debido a la incapacidad de aquel para codificar la información relativa al acceso sobre una red (de modo predeterminado, la mayor parte de las distribuciones de Linux se embarcan con Telnet desactivado y SSH activado).
3. *¿Necesita actualizaciones el software del servicio?* Todo software necesita actualizaciones de cuando en cuando, como los de los servidores Web y FTP. Esto se debe a que a medida que se agregan características, entran furtivamente nuevos problemas de seguridad. De modo que asegúrese de recordar que tiene que seguirle el rastro al desarrollo del software del servidor y tener las actualizaciones instaladas tan pronto como se publiquen los boletines relativos a la seguridad.

Interrupción de los servicios xinetd e inetd

Para interrumpir un servicio que se arrancó a través del programa `xinetd`, sencillamente edite el archivo de configuración de ese servicio en `/etc/xinetd` y dé el valor de Yes (Sí) a desactivar. Si está usando un `inetd` en existencia, edite el archivo `/etc/inetd.conf` y haga el comentario de que usted ya no lo quiere. Para designar el servicio como un comentario, inicie la línea con un signo de número (#). Vea el capítulo 8 para obtener más información sobre `xinetd` e `inetd`.

Recuerde enviar la señal HUP a `inetd`, una vez que haya hecho cualesquier cambios al archivo `/etc/inetd.conf` y una señal SIGUSR2 a `xinetd`. Si está usando Red Hat Linux o Fedora Core, también puede teclear el comando siguiente:

```
[root@hostA /root]# /etc/rc.d/init.d/xinetd reload
```

Interrupción de servicios que no sean inetd

Si un servicio no es ejecutado por **inetd**, entonces lo está ejecutando un proceso que es probable que se inicie en el momento de la inicialización. Si el servicio en cuestión fue instalado por su distribución y ésta ofrece una buena herramienta para desactivar un servicio, puede ser que encuentre que sea el procedimiento más sencillo.

Interrupción de servicios en Red Hat y Fedora Core Por ejemplo, bajo Fedora Core, RHEL y SuSE, el programa **chkconfig** proporciona una manera muy fácil para activar y desactivar servicios por separado. Por ejemplo, para desactivar el servicio **portmap**, sencillamente ejecute

```
[root@hostA ~]# chkconfig --level 35 portmap off
```

El parámetro **--level** se refiere a cuáles niveles de ejecución deben ser impactados por el cambio. Puesto que los niveles 3 y 5 representan los dos modos de usuarios múltiples, seleccionamos esos. El parámetro **portmap** es el nombre del servicio como se menciona en el directorio **/etc/rc.d/init.d**. Por último, el parámetro final puede ser “on” (activar), “off” (desactivar) o “reset” (restablecer). Las opciones “on” y “off” se explican por sí mismas. La opción “reset” se refiere a restablecer el servicio a su estado natural en el momento de la instalación.

Si quiere activar de nuevo el servicio **portmap**, sólo ejecute

```
[root@hostA ~]# chkconfig --level 35 portmap on
```

Note que el uso de **chkconfig** en realidad no activa o desactiva el servicio sino más bien define lo que sucederá en el momento del arranque. Para en realidad suspender el proceso, use el script **rc** del directorio **/etc/rc.d/init.d**. En el caso de **portmap**, lo suspenderíamos con

```
[root@hostA ~]# /etc/rc.d/init.d/portmap stop
```

Suspensión de los servicios en una manera independiente de la distribución Para impedir que un servicio arranque en el momento de la inicialización, cambie el symlink en el directorio **rc.d** del nivel correspondiente de ejecución. Esto se hace al ir al directorio **/etc/rc.d** y hallar en uno de los directorios **rc*.d** los symlinks que apunten al script de arranque (vea el capítulo 6 para obtener más información sobre los scripts de arranque). Renombré el symlink para empezar con una X en lugar de una S. Si decidiera reiniciar un servicio, es fácil renombrarlo de nuevo, empezando con una S. Si ha renombrado el script de arranque pero quiere detener el proceso actualmente en ejecución, use el comando **ps** para hallar el número ID del proceso y, después, comando **kill** para en realidad terminar el proceso. Por ejemplo, enseguida se dan los comandos para anular un proceso **portmap**, así como la salida resultante:

```
[root@hostA /root]# ps auxw | grep portmap
bin      255  0.0  0.1  1084  364 ?          S    Jul08   0:00 portmap
root     6634  0.0  0.1  1152  440 pts/0      S    01:55   0:00 grep portmap
[root@hostA /root]# kill 255
```

NOTA Como siempre, asegúrese de lo que está anulando, antes de anularlo, en especial en un servidor de producción.

MONITOREO DE SU SISTEMA

El proceso de atar la seguridad de su servidor no es sólo por consideración al aseguramiento de su sistema; le da la oportunidad de ver con claridad cómo debe verse el comportamiento normal de su servidor. Después de todo, una vez que conoce cuál es el comportamiento normal, el comportamiento desacostumbrado se sentirá como un dolor en el dedo pulgar (por ejemplo, si desactivó su servicio Telnet al hacer los ajustes en su servidor, ¡ver una entrada de registro para Telnet quiere decir que algo está mal!).

En realidad, existen paquetes comerciales que realizan monitoreo y puede ser que valga la pena hacer una comprobación para su sitio como un todo, pero dejaremos las discusiones de sus capacidades a *Network World* o a *PC Week*. Aquí echaremos un vistazo a una diversidad de otras excelentes herramientas que le ayudan a monitorear su sistema. Algunas de estas herramientas vienen con todas las distribuciones de Linux; algunas no. Todas son gratis y fáciles de adquirir.

Forma de hacer el mejor uso de syslog

En el capítulo 8 examinamos syslog, que realiza el registro cronológico en el sistema y que guarda los mensajes provenientes de varios programas en un conjunto de archivos de texto para los fines de mantener un registro. Por ahora, es probable que ya haya visto el tipo de mensajes de registro cronológico que obtiene con syslog. Éstos incluyen mensajes relacionados con la seguridad, como quién ha entrado al sistema, cuándo entró, etcétera.

Como puede imaginar, es posible analizar estos registros para tener una imagen durante un lapso de la utilización de los servicios de su sistema. Estos datos también pueden señalar actividad cuestionable. Por ejemplo, ¿por qué el anfitrión crackerboy.nothing-better-to-do.net (el-mañoso.nada-mejor-que-hacer.net) estuvo enviando tantas consultas Web en un periodo tan corto? ¿Qué estaba buscando? ¿Ha encontrado un agujero en el sistema?

Análisis del registro cronológico

La realización de verificaciones periódicas en los archivos de registro cronológico del sistema es una parte importante del mantenimiento de la seguridad. Por desgracia, examinar la enorme cantidad de registros de un día completo es una tarea tardada e infaliblemente aburrida que revela unos cuantos eventos significativos. Para facilitar la monótona labor, tome un texto en un lenguaje para escribir scripts (como Perl) y escriba pequeños scripts para analizar los registros. Un script bien diseñado funciona desembarazándose de lo que reconoce como comportamiento normal y mostrando todo lo demás. Esto puede reducir los miles de entradas de registro de una multitud de actividades de un día hasta unas cuantas docenas que se pueden manejar. Ésta es una manera eficaz para detectar intentos de intrusión y brechas posibles en la seguridad. Con buena suerte se volverá entretenido observar a los niñitos escritores de scripts tratando de echar abajo sus muros y fallando en el intento.

Almacenamiento de las entradas de registro cronológico

Por desgracia, el análisis de los registros puede ser insuficiente. Si alguien irrumpie en su sistema, es probable que de inmediato se borren sus archivos de registro, lo cual significa que todos esos maravillosos scripts no serán capaces de decirle algo. Para darle la vuelta a esto, considere dedicar un solo anfitrión en su red para almacenar sus entradas de registro. Configure su archivo `/etc/syslog.conf` para que envíe todos sus mensajes a este anfitrión único y configure éste de modo que sólo esté escuchando al puerto syslog (514). En la mayor parte de los casos, esto debe bastar para captar, en un lugar centralizado, la evidencia de cualesquiera cosas malas que estén sucediendo.

Si, *en realidad*, se está sintiendo paranoide, considere adjuntar una PC basada en DOS al puerto en serie del anfitrión de almacenamiento de los registros y, con el uso de paquete de emulación de terminal, como Telix, registre todos los mensajes enviados a este anfitrión (también puede usar otra caja Linux que ejecute **minicom** en el modo de registro cronológico; ¡sólo asegúrese de *no* conectar a la red esta segunda caja Linux!) Tenga **/etc/syslog.conf** configurado para enviar todos los mensajes a **/dev/ttyS0**, si está usando COM1, o a **/dev/ttyS1**, si está usando COM2. Y, por supuesto, *no* conecte el sistema DOS a la red. De esta manera, en el caso de que el anfitrión de registro también se vea atacado, los archivos correspondientes no serán destruidos. Los archivos de registro cronológico estarán seguros residiendo en el sistema DOS, al cual es imposible conectarse sin tener acceso físico.

Para obtener el grado más elevado de capacidad de monitoreo, conecte al sistema DOS una impresora en el puerto en paralelo y haga que el paquete de emulación de terminal repita todo lo que reciba en el puerto en serie de la impresora. De este modo, si el sistema DOS falla o resulta dañado de alguna manera por un ataque, tendrá una copia impresa de los registros (note que una desventaja seria de usar la impresora para realizar el registro cronológico es que no puede realizar una búsqueda con facilidad a lo largo de todos los registros. Si decide montar esta disposición, considere también guardar una copia electrónica para realizar las búsquedas con mayor facilidad).

SUGERENCIA Considere el uso de un paquete como **swatch** para que busque a usted cuando vea una entrada que indique problemas. Puede averiguar más acerca de este paquete en <http://swatch.sourceforge.net>.

Monitoreo del ancho de banda con MRTG

El monitoreo de la cantidad de ancho de banda que se está usando en sus servidores produce alguna información útil. El uso más práctico para ello es justificar la necesidad de actualizaciones. Al mostrar los niveles de utilización del sistema a los gerentes de usted, estará proporcionando números sólidos para respaldar sus afirmaciones. Sus datos también se pueden convertir con facilidad en una gráfica; ¡y a los gerentes les gustan las gráficas! Otro aspecto útil del monitoreo del ancho de banda es identificar cuellos de botella en el sistema, ayudando a usted de este modo a equilibrar mejor la carga de éste. Pero el aspecto más útil de trazar las gráficas de su ancho de banda es identificar cuándo van mal las cosas.

Una vez que ha instalado un paquete como MRTG (Multi-Router Traffic Grapher, Trazador de gráficas del tráfico de múltiples routers, el que se encuentra en <http://www.mrtg.org>) para monitorear el ancho de banda, rápidamente adquirirá un criterio respecto a cómo se ve lo “normal” en su sitio. Una caída o un incremento sustancial en la utilización es algo que investigar, ya que indica una falla o un tipo de ataque. Revise sus registros y vea sus archivos de configuración en búsqueda de entradas raras o desacostumbradas.

MANEJO DE LOS ATAQUES

Parte del aseguramiento de una red incluye la planeación para el peor de los casos: ¿qué sucede si alguien tiene éxito? No importa necesariamente de qué manera, sólo que ha sucedido. Los servidores están haciendo cosas que no deben hacer, se está fugando información que no debe fugarse, o usted, su equipo o alguien más descubre otra acción criminal, preguntándose por qué está tratando de difundir el pánico entre ellos.

¿Qué hace usted?

Precisamente como un director de instalaciones planea qué hacer en los incendios y su administrador de respaldos planea la recuperación de datos si ninguno de sus sistemas se encuentra dis-

ponible, un funcionario de seguridad necesita planear de qué manera manejar un ataque. En esta sección cubrimos los puntos que hay que considerar con respecto a Linux. Para obtener un excelente panorama general sobre el manejo de los ataques, visite el sitio Web CERT en <http://www.cert.org>.

En nada confíe

Si un atacante ha tenido éxito en sus sistemas, nada hay que puedan decirle sus servidores acerca de la situación que sea por completo confiable. Los “juegos de herramientas raíz”, o juegos de herramientas que usan los atacantes para invadir los sistemas y, a continuación, cubrir sus senderos, pueden dificultar la detección. Con los binarios reemplazados es posible hallar que nada puede hacer al propio servidor que sea de ayuda. En otras palabras, cada servidor que ha sido invadido con éxito por un hacker necesita reestructurarse por completo con una instalación nueva. Antes de realizar la reinstalación, haga un esfuerzo por mirar hacia atrás y saber cuán lejos llegó el atacante, como para determinar el punto en el ciclo de respaldo en donde se tiene la certeza que es confiable. Cualquier dato respaldado después de eso debe examinarse con todo cuidado, para garantizar que no se regresan datos no válidos al sistema.

Cambie sus contraseñas

Si el atacante ha obtenido su contraseña de raíz o quizás haya tomado una copia del archivo de contraseñas, resulta crucial que usted cambie sus contraseñas. Éste es un lío increíble; sin embargo, es necesario asegurarse de que el atacante no volverá a entrar bailando el vals a su servidor reestructurado usando la contraseña, sin resistencia alguna.

Note que es una buena idea cambiar también su contraseña de raíz si se tiene algún cambio en el personal. Puede parecer como que alguien está saliendo en buenos términos; no obstante, si averigua que alguien de su equipo tuvo problemas posteriores con la compañía podría significar que ya se encuentra usted también en problemas.

Suspensión del tráfico en la red

Una vez que está listo para iniciar la limpieza y necesita suspender todo acceso remoto al sistema, puede ser que encuentre necesario suspender todo el tráfico en la red hacia el servidor, hasta que se encuentre reestructurado por completo, con los parches más recientes, antes de volver a conectarlo a la red. Poner un servidor de regreso en la red cuando todavía se le están colocando parches es una manera casi segura de encontrarse de nuevo tratando con un ataque.

HERRAMIENTAS PARA LA SEGURIDAD EN LA RED

Existen incontables herramientas para ayudar a monitorear sus sistemas, incluyendo MRTG (<http://www.mrtg.org>) para trazar las gráficas de estadísticas (por lo general, valores SNMP provenientes de los conmutadores y routers), Big Brother (<http://www.bb4.org>) y, por supuesto, las diversas herramientas que ya hemos mencionado en este capítulo. Pero, ¿qué usa para meter las narices en su sistema con el fin de realizar verificaciones de sanidad básica?

En esta sección revisamos unas cuantas herramientas que puede usar para probar su sistema. Advierta que no basta una sola herramienta y ninguna combinación de herramientas es perfecta; no existe “Hackers Testing Toolkit” (Estuche de herramientas de prueba contra hackers) secreto que los profesionales de la seguridad tengan que nosotros no tengamos. La clave para cualquier herramienta es cómo la use, cómo interprete esos datos y tanto cuáles acciones conduzca al frente para proteger su sistema como las que tome después de obtener los datos de estas herramientas.

Una ilación común que encontrará en unas cuantas herramientas cuya lista se da enseguida es que, según el intento de sus diseñadores, no fueran concebidas como herramientas de seguridad. Lo que hace que esas herramientas funcionen bien para Linux desde una perspectiva de la seguridad es que ofrecen una visión más profunda de lo que está haciendo su sistema. Es esa información más profunda la que a menudo prueba ser de más ayuda de lo que puede usted haber pensado originalmente de ella.

nmap

El programa **nmap** escanea un anfitrión buscando puertos TCP y UDP abiertos. Cuando puede hallar uno, hace un intento de conexión de modo que puede identificar qué aplicación está activa en ese puerto. Ésta es una manera poderosa y sencilla para que un administrador eche una mirada a lo que su sistema expone a la red y la usan con frecuencia tanto los atacantes como los administradores para adquirir un sentido de lo que es posible contra un anfitrión.

Lo que hace poderoso a **nmap** es su capacidad para aplicar múltiples métodos de escaneo. Esto se hace porque cada método tiene sus pros y sus contras con respecto a qué tan bien recorre el firewall y tiene en cuenta el escaneo anónimo del anfitrión.

Snort

Un sistema de detección de intrusiones (IDS, *intrusion-detection system*) es una forma de monitorear de manera promiscua un punto en la red e informar sobre actividad cuestionable vista, con base en inspecciones de los paquetes. El programa Snort (<http://www.snort.org>) es una implementación de fuente abierta de un IDS que proporciona conjuntos extensos de reglas que se actualizan con frecuencia con nuevos vectores de ataque. Cualquier actividad cuestionable se registra en syslog y se cuenta con varias herramientas de fuente abierta de procesamiento de los registros (también en <http://www.snort.org>) para ayudar a hacer cabezas y colas de la información que está allí.

La ejecución de Snort en un sistema Linux que está colgando en un punto clave de entrada/salida de su red es una gran manera de seguir el rastro de la actividad, sin tener que estructurar un proxy para cada protocolo que desee soportar.

También existe una versión comercial de Snort llamada SourceFire. Puede averiguar más acerca de SourceFire en <http://www.sourcefire.com>.

Nessus

El sistema Nessus (<http://www.nessus.org>) toma la idea que se encuentra detrás de **nmap** y la extiende con sondas profundas del nivel de aplicación y una rica infraestructura de información. La ejecución de Nessus contra un servidor es una manera rápida de comprobación de la sanidad de la exposición de usted.

La clave para comprender Nessus es entender su salida. El informe registrará cronológicamente numerosos comentarios, desde un nivel de información en todo el camino hasta un nivel elevado. Dependiendo de cómo se escriba la aplicación de usted y de cuáles otros servicios ofrece en su sistema Linux, Nessus puede registrar positivos falsos o notas espeluznantes de información. Tome el tiempo para leer por completo cada una de ellas y entienda lo que es la salida, ya que no todos los mensajes reflejan de manera necesaria su situación. Por ejemplo, si Nessus detecta que su sistema se encuentra en riesgo debido a un agujero en Oracle 8, pero su sistema Linux no ejecuta éste, lo más probable es que se halla usted topado con un positivo falso.

Aun cuando Nessus es de fuente abierta y gratis, es propiedad de una compañía comercial, Tenable Network Security, la cual lo administra. Puede aprender más acerca de Tenable en <http://www.tenablesecurity.com>.

Ethereal/tcpdump

En el capítulo 11 aprendimos de manera extensa acerca de Ethereal y **tcpdump**, en donde las usamos para estudiar las entradas y salidas de TCP/IP. Aun cuando sólo hemos visto estas herramientas usadas para detección de fallas, son precisamente tan valiosas para realizar funciones de seguridad de la red.

Las inspecciones en bruto de la red son el alimento que todas las herramientas cuya lista se da en las secciones anteriores toman para adquirir una visión de lo que está haciendo su servidor. Sin embargo, no tienen del todo la visión de lo que se *supone* que usted hace que su servidor haga. Por consiguiente, resulta útil que usted mismo sea capaz de tomar inspecciones de la red y leer de uno a otro lado de ellas para ver si hay alguna actividad cuestionable que se esté llevando a efecto. ¡Puede ser que se sorprenda de lo que su servidor está haciendo!

Por ejemplo, si está mirando una posible intrusión, puede ser que quiera iniciar una inspección en bruto de la red desde otro sistema Linux que pueda ver todo el tráfico de la red de su anfitrión cuestionado. Mediante la captura de todo el tráfico durante un periodo de 24 horas, puede usted regresar y aplicar filtros para ver si existe algo que no debe estar allí. Extendiendo el ejemplo, si se supone que el servidor sólo maneja operaciones Web y SSH con la resolución DNS inversa desactivada en ambas, tome la inspección y aplique el filtro “not port 80 and not port 22 and not icmp and not arp” (ningún puerto 80 y ningún puerto 22 y ningún icmp y ningún arp). Cualesquiera paquetes que se muestren en la salida son sospechosos.

RESUMEN

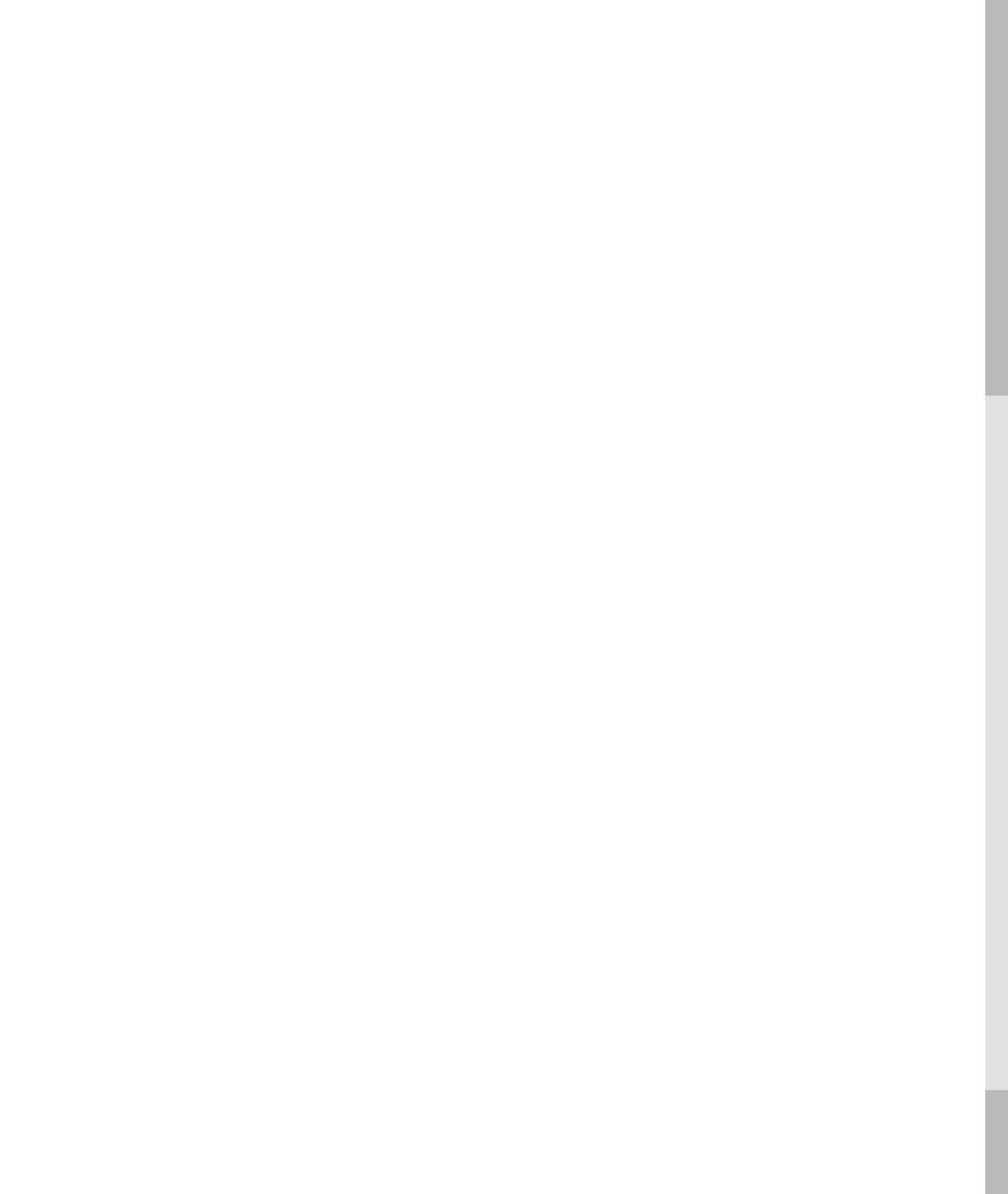
En este capítulo cubrimos los aspectos básicos de la seguridad en la red como pertenece a Linux. Con la información que se da aquí, debe tener la información que necesita para tomar una decisión informada acerca del estado de salud de su servidor y decidir cuál acción (si la hay) es necesaria para asegurarlo mejor.

Como se ha indicado en otros capítulos, por favor no considere éste como una fuente completa de información sobre seguridad en la red. La seguridad, como un campo, está evolucionando de manera constante y requiere un ojo cuidadoso hacia lo que es nuevo. Asegúrese de suscribirse a las listas pertinentes de correos, lea los sitios Web y, si es necesario, elija un libro como *Network Security, A Beginners Guide* por Eric Maiwald (McGraw-Hill / Osborne, 2003).

PARTE IV



Servicios
de Internet



CAPÍTULO 16



DNS

La necesidad de mapear direcciones IP numéricas no amigables en un formato más amigable ha sido un propósito de TCP/IP desde su creación en la década de 1970. Aunque dicha traducción no es obligatoria, sí hace que la red sea mucho más útil y fácil de manejar para las personas.

En un inicio el mapeo entre el nombre y la dirección IP se hacía mediante el mantenimiento a un archivo **hosts.txt** que se distribuía vía FTP a todas las máquinas en Internet. Conforme creció el número de anfitriones (lo cual sucedió a inicios de la década de 1980), pronto fue claro que una sola persona dedicada a mantener un solo archivo para todos los anfitriones no era una forma escalable de administrar la asociación entre direcciones IP y nombres de anfitriones. Para resolver este problema, se ideó un sistema distribuido en el cual cada sitio mantendría información acerca de sus propios anfitriones. Un anfitrión en cada sitio se consideraría como “autorizado”, y esa única dirección del anfitrión se guardaría en una tabla maestra que podría ser consultada por el resto de los sitios. Ésta es la esencia del *Domain Name Service (DNS)* (*Servicio de Nombres de Dominio*).

Si la información del DNS no estuviera descentralizada como ahora, otra opción hubiera sido mantener en un sitio central una lista maestra de todos los anfitriones (lo cual asciende a decenas de millones) y su actualización hubiera tenido que realizarse decenas de miles de veces al día, ¡esta alternativa pronto se habría convertido en una labor titánica! Pero las necesidades de cada sitio son más importantes. Un sitio puede necesitar de un servidor DNS privado debido a que su firewall requiere direcciones IP para su LAN que no sean visibles para las redes en el exterior, aunque los anfitriones en dicha LAN deben ser capaces de encontrar otros anfitriones en Internet. Si usted se asombra con la perspectiva de tener que administrar esto para cada anfitrión en Internet, entonces comienza a tener idea de lo que hablamos.

NOTA En este capítulo usted verá que los términos “servidor DNS” y “servidor de nombres” se usan de manera intercambiable. En rigor técnico, “servidor de nombres” es algo ambiguo porque puede aplicarse a cualquier número de esquemas de nombramiento que traduzcan un nombre en un número, y viceversa. Sin embargo, en el contexto de este capítulo, “servidor de nombres” siempre significará servidor DNS a menos que se indique lo contrario.

Discutiremos DNS a fondo, de manera que usted tenga lo que necesita a fin de configurar y desplegar sus propios servidores DNS para cualesquiera necesidades que tenga.

EL ARCHIVO DE ANFITRIONES

No todos los sitios tienen sus propios servidores DNS. Ni todos los sitios necesitan sus propios servidores DNS. En sitios suficientemente pequeños sin conexión a Internet es razonable para cada anfitrión mantener su propia copia de una lista que asocie todos los nombres de anfitrión en la red local con sus direcciones IP correspondientes. En la mayoría de los sistemas Linux y UNIX, esta tabla se almacena en el archivo **/etc/hosts**.

NOTA Aun en anfitriones que tienen acceso a un servidor DNS puede haber razones válidas para mantener un archivo de anfitriones de manera local. Ello es útil para que un anfitrión pueda buscar una dirección IP en forma local antes de salir a preguntarle a un servidor DNS. Por lo general, esto se hace para que el sistema pueda ubicar los anfitriones que necesita para el arranque de manera que aun si el servidor DNS no está disponible, el sistema puede iniciar sin problemas. Menos obvia puede ser la sencilla razón de que usted quiere asignar un nombre a un anfitrión pero no quiere (o no puede) añadir esa entrada al servidor DNS.

El archivo `/etc/hosts` mantiene su información en un formato tabular simple y es un servicio de nombramiento básico. La dirección IP aparece en la primera columna y todos los nombres de anfitriones relacionados en la segunda columna. La tercera columna se utiliza comúnmente para guardar una versión corta del nombre del anfitrión. Sólo espacio blanco separa los campos. El símbolo de número (#) al principio de un renglón representa comentarios. He aquí un ejemplo:

```
# Tabla de anfitriones para la red interna
#
127.0.0.1      localhost.localdomain  localhost
192.168.1.1    servidorA.ejemplo.org  serverA      # Servidor Linux
192.168.1.2    servidorB.ejemplo.org  serverB      # Otro servidor Linux
192.168.1.7    dikkog                  # Servidor Win2003
192.168.1.8    trillian                # Nodo maestro del clúster
192.168.1.9    sassy                   # Caja FreeBSD
10.0.88.20     laserjet5               # Impresora del comedor
```

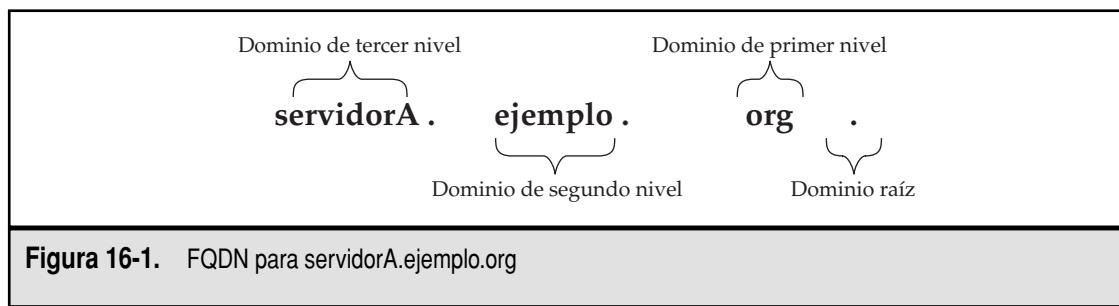
En general, su archivo `/etc/hosts` debe contener cuando menos los mapeos IP-anfitrión necesarios para la interfaz de retorno (127.0.0.1) y el nombre del anfitrión local con la correspondiente dirección IP. Un sistema DNS es un servicio de nombramiento más robusto. El resto del capítulo cubrirá el uso del servicio de nombres DNS.

PRIMERO, ENTENDER CÓMO TRABAJA DNS

En esta sección exploraremos algo de material previo necesario para el entendimiento de la instalación y la configuración de un servidor DNS y un cliente.

Dominios y convencionalismos sobre nombramiento de anfitriones

Hasta ahora, lo más seguro es que usted esté haciendo referencia a sitios por su *Fully Qualified Domain Name (FQDN)* (*Nombre de dominio completamente calificado*) como este: `www.kernel.org`. Cada cadena entre los puntos en este FQDN es significativa. Empezando por la derecha y moviéndose hacia la izquierda, podrá encontrar el primer componente de nivel de dominio, luego el segundo y finalmente el tercero. Esto se ilustra mejor en la figura 16-1, en el FQDN para un sistema (`servidorA.ejemplo.org`) y es un ejemplo clásico de un FQDN. Su división se analiza en detalle en la siguiente sección.



El dominio raíz

La estructura del DNS se asemeja a un árbol invertido (árbol de cabeza); por lo tanto, esto significa que la raíz está hacia arriba dejando a las ramas ¡hacia abajo! Simpático arbólito, ¿no le parece?

En la cima del árbol invertido de dominios está el máximo nivel de la estructura DNS, acertadamente llamada dominio raíz y representada por un simple punto (.).

Éste es el punto que se supone debería aparecer al final de todo FQDN pero que se supone en silencio que está presente aunque explícitamente no va escrito. Así, por ejemplo, el FQDN apropiado para www.kernel.org es realmente www.kernel.org. (con el punto raíz al final). Y el FQDN para el afamado portal Web de Yahoo! es en realidad www.yahoo.com. (de igual forma).

No es coincidencia entonces que esta porción del espacio de nombres de dominio es administrada por un grupo de servidores especiales conocidos como los *servidores de nombre raíz*. Al momento de escribir este libro, había un total de 13 servidores de nombre raíz administrados por 13 proveedores (y cada proveedor puede tener múltiples servidores que están dispersos alrededor del mundo. Los servidores están dispersos por varias razones, tales como seguridad y balanceo de cargas). Los servidores de nombre raíz se nombran en orden alfabético. Tienen nombres como a.root-server.net, b.root-server.net, ... m.root-server.net. El papel de estos servidores de nombre raíz será abordado más adelante.

Los nombres de dominio de primer nivel

Los Top-Level Domains (TLD) (Nombres de dominio de primer nivel) se pueden considerar como las primeras ramas que crecerían hacia abajo en nuestra estructura de árbol invertido.

Podríamos hacer una suposición descabellada y decir que los dominios de primer nivel proporcionan la organización categórica del espacio de nombres DNS. Lo que esto significa en español llano es que las diversas ramas del espacio de nombres de dominio se han dividido en categorías claras para ajustarse a diferentes usos (ejemplos de tales usos serían geográficos, funcionales, etc.). Al momento de escribir esta obra había aproximadamente unos 258 dominios de primer nivel.

Los TLD pueden subdividirse aún más en dominios genéricos de primer nivel (por ejemplo, .org, .com, .net, .mil, .gov, .edu, .int, .biz), dominios de primer nivel para códigos de país (por ejemplo, .us, .uk, .ng y .ca, que corresponden respectivamente a los códigos de país para Estados Unidos, el Reino Unido, Nigeria y Canadá), y otros dominios especiales de primer nivel (por ejemplo el dominio .arpa).

El dominio de primer nivel en nuestro ejemplo FQDN (servidorA.ejemplo.org) es “.org”.

Los nombres de dominio de segundo nivel

Los nombres en este nivel del DNS conforman el límite organizacional presente en el espacio de nombres. Compañías, Internet Service Providers (ISP) (Proveedores de servicios de Internet) comunidades educativas, organizaciones sin fines de lucro e individuos adquieren típicamente nombres únicos en este nivel. He aquí algunos ejemplos: redhat.com, caldera.com, planetoid.org, labmanual.org, kernel.org.

El dominio de segundo nivel en nuestro ejemplo FQDN (servidorA.ejemplo.org) es “.ejemplo”.

Los nombres de dominio de tercer nivel

A este nivel del espacio de nombres de dominio, individuos y organizaciones a los que se les ha asignado un nombre de dominio de segundo nivel pueden decidir con cierta libertad qué hacer con los nombres de tercer nivel. Lo convencional, sin embargo, es utilizar los nombres de tercer

nivel para colocar nombres de anfitriones u otros usos funcionales. También es común que las organizaciones inicien definiciones de subdominios en este nivel. Un ejemplo de la asignación funcional de un nombre de dominio de tercer nivel será la “www” en el FQDN www.yahoo.com. La “www” aquí puede ser el nombre de una máquina anfitrión debajo del entorno definido por el dominio yahoo.com, o bien, puede ser un alias hacia el nombre de un anfitrión existente.

El nombre de tercer nivel en nuestro FQDN de ejemplo (servidorA.ejemplo.org) es “servidorA”. No representa otra cosa que el nombre de un anfitrión existente en nuestro sistema.

Al mantener el DNS distribuido de esta manera, la tarea de mantener el control de todos los anfitriones conectados a Internet se delega a cada sitio que asume el cuidado de su propia información. El almacén central de listados de todos aquellos nombres de servidores primarios, llamados los *servidores raíz*, es la única lista de dominios existentes. Desde luego, una lista de naturaleza tan crítica está duplicada en varios servidores y diversas regiones geográficas. Por ejemplo, un terremoto en Japón podría destruir el servidor raíz de Asia pero el resto de los servidores raíz ubicados alrededor del mundo pueden suplir su ausencia en tanto se restablece el servidor. La única diferencia notable para los usuarios sería un ligero incremento de latencia al resolver nombres de dominio. Asombroso, ¿no cree? La estructura del árbol invertido del DNS se muestra en la figura 16-2.

Subdominios

“¡Pero yo acabo de ver el sitio www.soporte.ejemplo.org!”, dirá usted. “¿Cuál es el componente del nombre de dominio y cuál es el componente del nombre del anfitrión?”

Bienvenido al turbulento y misterioso mundo de los *subdominios*. Un subdominio exhibe todas las propiedades de un dominio, excepto que ha delegado una subsección del dominio en vez de todos los anfitriones en un sitio. Utilizando el sitio ejemplo.org como ejemplo, el subdominio para el departamento de soporte y mesa de ayuda de Ejemplo, S.A. de C.V., es soporte.ejemplo.org. Cuando el servidor de nombres primario para el dominio ejemplo.org recibe una solicitud por un nombre de anfitrión cuyo FQDN termina en soporte.ejemplo.org, el servidor primario redirige la solicitud hacia el servidor de nombres primario de soporte.ejemplo.org. Sólo el servidor de nombres primario de soporte.ejemplo.org conoce todos aquellos anfitriones debajo de él, anfitriones tales como un sistema llamado “www” con el FQDN “www.soporte.ejemplo.org”.

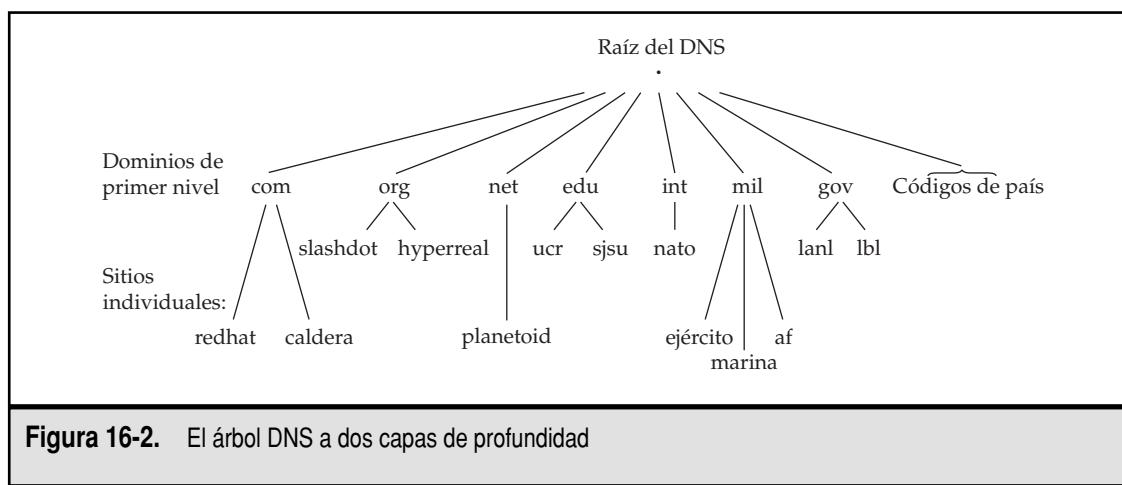


Figura 16-2. El árbol DNS a dos capas de profundidad

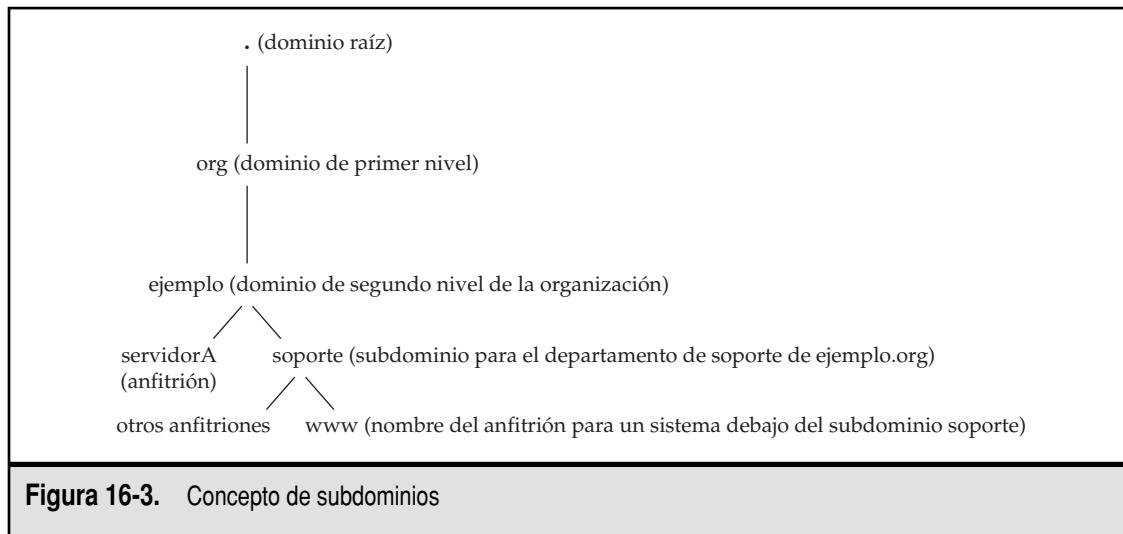


Figura 16-3. Concepto de subdominios

La figura 16-3 muestra la relación desde los servidores raíz bajando hasta ejemplo.org y luego hacia soporte.ejemplo.org. La “www” es, claro está, el nombre de un anfitrión.

Para hacer esto aún más claro, sigamos la ruta de una solicitud DNS:

1. Un cliente quiere visitar un sitio Web llamado “www.soporte.ejemplo.org”.
2. La solicitud inicia con el dominio de primer nivel “org”. Dentro de “org” está “ejemplo.org”.
3. Digamos que uno de los servidores DNS que tiene la autoridad para el dominio “ejemplo.org” se llama “ns1.ejemplo.org”.
4. Ya que el anfitrión ns1 es el que tiene la autoridad para el dominio ejemplo.org, debemos preguntarle por todos los anfitriones (y subdominios) debajo de él.
5. Así que le preguntamos por información acerca del anfitrión sobre el cual estamos interesados: “www.soporte.ejemplo.org”.
6. La configuración DNS de ns1.ejemplo.org es tal que para cualquier solicitud que termine en soporte.ejemplo.org el servidor contactará a otro servidor con autoridad llamado “dns2.ejemplo.org”.
7. La solicitud de “www.soporte.ejemplo.org” se pasa entonces a dns2.ejemplo.org, el cual regresa la dirección IP para www.soporte.ejemplo.org, digamos, 192.168.1.10.

Note que, cuando un nombre de un sitio parece reflejar la presencia de subdominios, no necesariamente significa que en realidad existan. Aunque las reglas de especificación de nombres de anfitrión no permiten puntos, el servidor de nombres del Berkeley Internet Name Domain (BIND) (Dominio de nombres de Internet de Berkeley) siempre los ha permitido. Así, de vez en cuando, usted verá puntos en los nombres de anfitrión. Que exista o no un subdominio lo maneja la configuración del servidor DNS para el sitio. Por ejemplo, www.falso.ejemplo.org no implica en forma predeterminada que falso.ejemplo.org es un subdominio. En vez de ello, quizás signifique que “www.falso” es el nombre de anfitrión para un sistema dentro del dominio ejemplo.org.

El dominio in-addr.arpa

El DNS permite a la resolución trabajar en ambos sentidos. La *resolución hacia delante* convierte nombres en direcciones IP, y la *resolución inversa* convierte direcciones IP de vuelta en nombres de anfitriones. El proceso de resolución inversa depende del dominio *in-addr.arpa*, donde arpa es un acrónimo para Address Routing and Parameters Area (Área de Parámetros y Enrutamiento de Direcciones).

Como se explicó en la sección anterior, los nombres de dominio son resueltos al examinar cada componente de derecha a izquierda, con el punto como sufijo que indica la raíz del árbol DNS. Siguiendo esta lógica, las direcciones IP también deben tener un dominio de primer nivel. Este dominio se llama *in-addr.arpa*.

A diferencia de los FQDN, las direcciones IP se determinan de izquierda a derecha una vez que están debajo del dominio *in-addr.arpa*. Cada octeto reduce aún más los posibles nombres de anfitrión. La figura 16-4 proporciona un ejemplo visual de la resolución inversa para la dirección IP 138.23.169.15.

Tipos de servidores

Los servidores DNS vienen en tres maneras: primarios, secundarios y de caché. Otra clase especial de servidores de nombres consiste en los llamados “servidores de nombre raíz”. Otros servidores DNS requieren de vez en cuando el servicio proporcionado por los servidores de nombre raíz.

Los tres principales servidores DNS se analizan a continuación:

Servidores primarios son aquellos que se considera que tienen la autoridad para un dominio en particular. Un *servidor que tiene la autoridad* es aquel en donde residen los archivos de configuración del dominio. Cuando ocurren actualizaciones a las tablas DNS del dominio, se efectúan en estos servidores. Un servidor de nombres primario no es otra cosa que un servidor DNS que sabe acerca de todos los anfitriones y subdominios que existen bajo el dominio que atiende.

Servidores secundarios son aquellos que sirven de respaldo y distribuidores de carga para los servidores de nombres primarios. Los servidores primarios saben de la existencia de los secundarios y les mandan actualizaciones periódicas para las tablas de nombres. Cuando un sitio hace una solicitud a un servidor de nombres secundario, éste responde en forma autorizada. Sin embargo, como puede suceder que el secundario reciba una solicitud antes de que el primario le alerte sobre los últimos cambios, algunas personas llaman a los secundarios “nada autorizados”. En realidad, es posible confiar en que los secundarios tendrán la información correcta (además, a menos que sepa cuál es cuál, no podrá notar diferencia alguna entre una solicitud respondida por un primario y una recibida de un secundario).

Servidores de nombre raíz

Los servidores de nombre raíz operan como el primer puerto de llamada para las partes más elevadas del espacio de nombres de dominio. Estos servidores publican un archivo llamado “archivo de zona raíz” para otros servidores DNS y clientes en Internet. El archivo de zona raíz describe dónde se localizan los servidores que tienen la autoridad para los dominios primarios del DNS (com, org, ca, ng, hk, uk, etc.).

Un servidor de nombre raíz es sólo una instancia de un servidor de nombres primario, sólo delega toda solicitud que recibe hacia otro servidor de nombres. Usted puede construir su propio servidor raíz fuera del BIND, ¡no hay nada especial en ello!

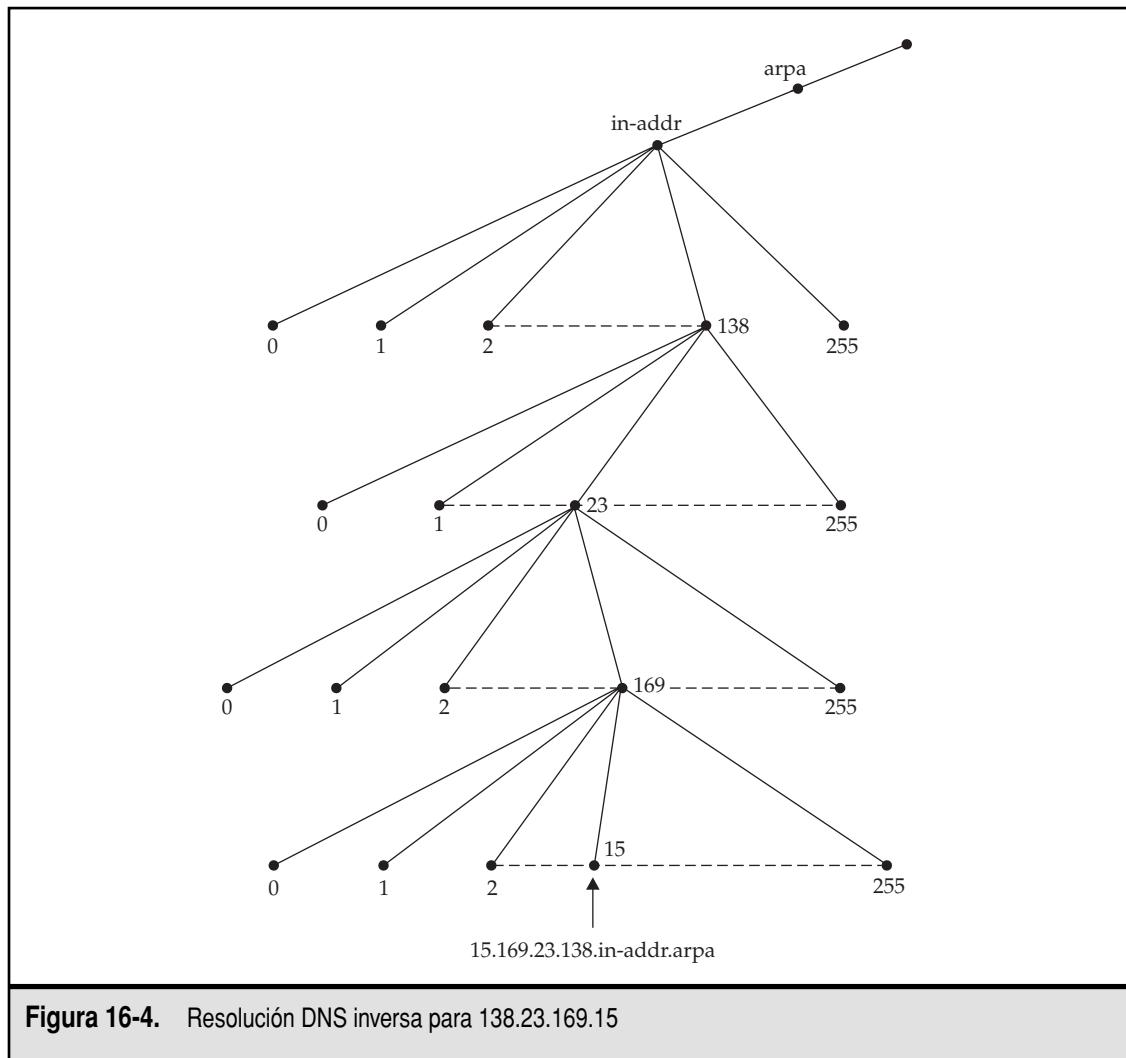


Figura 16-4. Resolución DNS inversa para 138.23.169.15

Servidores de caché: son eso, servidores de caché o de almacenamiento. No contienen archivos de configuración para ningún dominio en particular. En vez de ello, cuando un anfitrión cliente solicita a un servidor de caché que determine un nombre, tal servidor verificará primero su propia memoria. Si no puede encontrar una coincidencia, encontrará al servidor primario para preguntarle. Acto seguido, la respuesta se almacena. En términos prácticos, los servidores de caché trabajan bastante bien debido a la naturaleza temporal de las solicitudes de DNS. Esto es, si se solicita la dirección IP para hyperreal.org, es probable que ésta se vuelva a solicitar en un futuro cercano (la Web ha hecho esto más probable todavía). Los clientes pueden establecer la diferencia entre un servidor de caché y uno primario o uno secundario. Ello se debe a que, cuando un servidor de caché responde una solicitud, lo hace de manera “no autorizada”.

NOTA Un servidor DNS se puede configurar para actuar con cierto nivel de autoridad para un dominio en particular. Por ejemplo, un servidor puede ser primario para ejemplo.org pero secundario para dominio.com. Todos los servidores DNS actúan como servidores de caché, sin importar si son primarios o secundarios para cualesquier otros dominios.

INSTALACIÓN DE UN SERVIDOR DNS

No hay mucha variedad en el software disponible para un servidor DNS, pero existen dos tipos en particular dentro del mundo de Linux/UNIX: djbdns y el venerable servidor Berkeley Internet Name Domain (BIND) (Dominio de nombres de Internet de Berkeley). djbdns es una solución de DNS ligera cuyos creadores claman que es un reemplazo más seguro del BIND. Este último es el veterano y, por ende, es un programa de sobrada popularidad. Se usa en un vasto número de máquinas que sirven nombres alrededor del mundo. Hoy, BIND lo desarrolla y mantiene el ISC, Internet Systems Consortium, Consorcio de sistemas de Internet. Para obtener más información sobre ISC visite <http://www.isc.org/>. El ISC está a cargo del desarrollo del ISC DHCP cliente/servidor entre otros programas.

Debido al tiempo que tomó escribir este libro y la inevitable liberación de nuevas versiones del software, es posible que la versión del BIND presentada aquí no sea la misma versión a la que tenga acceso; sin embargo, no debe preocuparse por ello en absoluto, pues muchas de las directrices de configuración, palabras clave y sintaxis de comandos se han mantenido sin cambios en buena medida entre las versiones recientes de este software.

Nuestro sistema de ejemplo se ejecuta en la versión Fedora Core de Linux, y como tal, estaremos utilizando el binario precompilado que se envía con este sistema operativo. Se supone que el software enviado con Fedora es reciente, así que puede estar seguro de que la versión del BIND referida aquí es cercana a la última versión que puede ser obtenida directamente del sitio www.isc.org (en dicho sitio podrá encontrar RPM precompilados del programa BIND).

La buena noticia es que, una vez que BIND quede configurado, será raro que necesite preocuparse por su operación. Sin embargo, recuerde la importancia de buscar nuevas versiones. De vez en cuando se descubren fallas y cuestiones de seguridad y deben ser enmendadas instalando la nueva versión. Desde luego, también se añaden nuevas características pero, a menos que las necesite, ese tipo de actualizaciones pueden esperar.

El programa BIND se puede encontrar en el directorio **/Fedora/RPMS/** en la raíz del DVD de Fedora. También lo puede descargar a su sistema local desde cualquiera de los espejos de Fedora. Visite http://download.fedoraproject.org/pub/fedora/linux/core/4/i386/os/Fedora/RPMS/bind-9*.rpm.

Suponiendo que descargó o copió los binarios del BIND en su directorio de trabajo, puede instalarlo utilizando el comando **rpm**. Escriba

```
[root@serverA root]# rpm -Uvh bind-9*
```

Si tiene una conexión a Internet, la instalación de BIND puede ser tan sencilla como escribir el siguiente comando:

```
[root@serverA root]# up2date -i bind
```

Una vez que este comando finalice, estará listo para iniciar la configuración del servidor y ejecutar el servidor DNS.

Descarga, compilación e instalación del software ISC BIND desde la fuente

Si el software ISC BIND no viene incluido en forma predeterminada en la distribución Linux que tenga disponible, puede armar el software a partir del código fuente disponible en el sitio del ISC ubicado en <http://www.isc.org>. También es posible que nadie más quiera aprovechar las más recientes enmiendas que están disponibles para el software, lo que su paquete de distribución quizás no incorpora todavía. Al momento de escribir este libro, la versión más estable y actualizada del software era la versión 9.3.1, la cual puede ser descargada en forma directa del <ftp://ftp.isc.org/isc/bind9/9.3.1/bind-9.3.1.tar.gz>.

Una vez que el paquete ha sido descargado, desempáquelo como se muestra enseguida. Para este ejemplo, suponemos que la fuente se descargó en el directorio **/usr/local/src/**. Desempaque pues el archivo tar de la siguiente manera:

```
[root@serverA src]# tar xvzf bind-9.3.1.tar.gz
```

Entre al directorio **bind*** creado por el comando anterior y luego destine un minuto para estudiar cualesquier archivos README (LÉAME) disponibles.

Enseguida configure el paquete con el comando **configure**. Suponiendo que deseamos que BIND quede instalado en el directorio **/usr/local/named**, emplearemos

```
[root@serverA bind-9.3.1]# ./configure --prefix=/usr/local/named
```

Cree el directorio especificado por la opción “prefix” utilizando **mkdir**:

```
[root@serverA bind-9.3.1]# mkdir /usr/local/named
```

Para compilar e instalar, utilice los comandos **make**; **make install**:

```
[root@serverA bind-9.3.1]# make ; make install
```

La versión del software ISC BIND que acabamos de construir desde la fuente instala el programa residente del servidor de nombres (**named**) y otras convenientes utilidades dentro del directorio **/usr/local/named/sbin/**. Los programas para los clientes (**dig**, **oct**, **nsupdate**, etc.) se instalan dentro del directorio **/usr/local/named/bin/**.

Lo que se instaló

Muchos programas vienen con el paquete principal del **bind** y con el paquete **bind-utils** que se instalaron. Las cuatro herramientas que nos interesan son las siguientes:

Herramienta	Descripción
/usr/sbin/named	El programa DNS en sí
/usr/sbin/rndc	La herramienta de control del servidor de nombres bind
/usr/bin/host	Realiza una solicitud sencilla en un servidor de nombres
/usr/bin/dig	Ejecuta solicitudes complejas en un servidor de nombres

El resto del capítulo tratará algunos de los programas / utilidades listadas aquí, al igual que su configuración y su uso.

Comprensión del archivo de configuración BIND

El archivo **named.conf** es el archivo principal de configuración para BIND. Basándose en las especificaciones contenidas en este archivo, BIND determina cómo debe comportarse y qué otros archivos de configuración tiene que leer, si los hubiere.

La presente sección de este capítulo cubre lo que es necesario saber para poner en funcionamiento un servidor DNS de propósito general. Una guía completa sobre el nuevo formato del archivo de configuración está ubicada dentro del directorio **html** de la documentación del BIND.

El formato general del archivo **named.conf** es como sigue:

```
statement {
    opciones;      // comentarios
};
```

El **statement** (enunciado) le dice a BIND que está a punto de describir una faceta particular de su operación mientras que las **opciones** son los comandos específicos que aplican al enunciado. Se necesitan los signos de llave a fin de que BIND determine qué opciones están relacionadas con los enunciados a las que corresponden. Hay un punto y coma después de cada opción y de la llave que cierra.

Mostramos un ejemplo de esto a continuación:

```
options {
    directory "/var/named";   // pone los archivos de configuración en /var/named
};
```

El enunciado **bind** que precede significa que éste es un enunciado opcional, y que la opción en particular es una directriz que especifica el directorio de trabajo para **bind**, esto es, el directorio en el sistema de archivos local que almacenará los datos de configuración del servidor de nombres.

Los detalles

Esta sección documenta los enunciados más comunes que pueden verse en el archivo **named.conf**. La mejor manera de enfrentar esto es darle sólo un vistazo pero mantenerlo como referencia para secciones posteriores del libro. Si alguna de las directrices parece extraña o no parece tener sentido en el primer vistazo, no se preocupe. Una vez que las vea en secciones posteriores, los cómo y los por qué caerán en su sitio.

Comentarios

Los comentarios pueden escribirse en alguno de los siguientes formatos:

Formato	Indica
//	Comentarios al estilo C++
/*...*/	Comentarios al estilo C
#	Comentarios al estilo script de shell Perl y UNIX

En el caso del primer y el tercer estilo (C++ y Perl/UNIX), una vez que el comentario empieza, éste continúa hasta el final de la línea. En los comentarios estilo C regular, se requiere */ para señalar el fin del comentario. Esto hace que el estilo C facilite la inserción de un comentario de varios renglones. Sin embargo, en términos generales, puede escoger el formato de comentarios que mejor le parezca y quedarse con ese. No hay un estilo mejor que otro.

Palabras reservadas en el enunciado

Puede utilizar las siguientes palabras reservadas:

Palabra reservada	Descripción
acl	Lista de control de acceso (Access Control List) que determina qué tipo de acceso tendrán otros a su servidor DNS.
include	Le permite incluir otro archivo y hacer que éste sea tratado como parte del archivo named.conf normal.
logging	Especifica qué información se registrará y cuál será ignorada. Para la información registrada, también puede especificar dónde se registrará.
options	Aborda esquemas de configuración global del servidor.
controls	Permite declarar canales de control a ser utilizados por la utilidad rndc .
server	Ajusta opciones de configuración específicas para el servidor.
zone	Define la zona DNS.

El enunciado include

Si observa que su archivo de configuración comienza a crecer fuera de control, quizás quiera dividirlo en varios archivos pequeños. Cada archivo puede ser incluido en el archivo **named.conf** principal. Note que no puede utilizar el enunciado **include** dentro de otro enunciado.

He aquí un ejemplo del enunciado **include**:

```
include "/ruta/al/archivo_a_ser_incluido";
```

NOTA Para todos aquellos programadores en C y C++: asegúrense de no iniciar las líneas `include` con el símbolo de número (#), ¡sin importar lo que su instinto les ordene! Ese mismo símbolo se usa para iniciar comentarios en el archivo `named.conf` file.

Enunciado registro cronológico (logging)

Este enunciado se utiliza para especificar qué información quedará registrada, y dónde. Cuando este enunciado se usa en conjunción con la instalación `syslog`, obtiene un sistema de registro cronológico extremadamente poderoso y configurable. Los artículos registrados son una cierta cantidad de estadísticas acerca del estado de `named`. En forma predeterminada, se registran en el archivo `/var/log/messages`. En su forma más simple, los diversos tipos de registros han agrupado en categorías predefinidas; por ejemplo, hay categorías para registros relacionados con la *seguridad (security)*, una categoría *general (general)*, una categoría *predeterminada (default)*, una categoría de *determinador (resolver)*, otra de *solicitudes (queries)*, etcétera.

Desafortunadamente, la posibilidad de configurar el enunciado registro cronológico viene con el precio de la complejidad adicional, pero las configuraciones predeterminadas para los registros de `named` son suficientemente buenas para la mayoría de los casos. He aquí un ejemplo de una sencilla directriz de registros cronológicos:

```
1      logging {  
2          category default { default_syslog; } ;  
3          category queries { default_syslog; } ;  
4      };  
5
```

NOTA Se añaden números de línea para facilitar la lectura del listado.

La especificación de registro cronológico precedente significa que todos los registros que caen dentro de la categoría predeterminada (default) se enviarán al registro syslog del sistema (la categoría predeterminada define las opciones de registro cronológicos para aquellas categorías donde ninguna configuración específica se ha definido).

La línea 3 en el listado anterior especifica dónde se registrarán todas las solicitudes; en este caso, todas ellas se registrarán en el syslog del sistema.

Enunciado server

Este enunciado proporciona al BIND información específica acerca de otros servidores de nombres con los cuales podría lidiar. El formato del enunciado `server` es el siguiente:

```
1      server dirección-ip {  
2          bogus yes/no;  
3          keys { string ; [ string ; [...] ] } ;  
4          transfer-format one-answer/many-answers;  
5          ...<otras opciones>...  
6      };
```

donde `ip-address` en la línea 1 es la dirección IP del servidor de nombres remoto en cuestión.

La opción **bogus** en la línea 2 le dice al servidor si el servidor remoto está enviando información errónea. Ello es útil en caso de que esté lidiando con otro sitio que esté enviando información no confiable debido a una mala configuración. La cláusula **keys** en la línea 3 especifica un **key_id** (identificador de llave) que se define en el enunciado **key**, el cual puede ser utilizado para asegurar transacciones cuando se comunica con un servidor remoto. Esta llave se usa al generar una firma de requisición que se añade al final de los mensajes intercambiados con el servidor remoto. El comando de la línea 4, **transfer-format**, informa al BIND si el servidor remoto puede aceptar respuestas múltiples en la respuesta a una sola solicitud.

Un ejemplo del enunciado **server** podría verse como esto:

```
server 192.168.1.12 {  
    bogus no;  
    transfer-format many-answers;  
};
```

Zonas

Este enunciado le permite definir una zona DNS, cuya definición es con frecuencia engañosa. He aquí la letra pequeña: *una zona DNS no es lo mismo que un dominio DNS*. La diferencia es sutil pero importante.

Repasemos: los dominios son designados junto con los límites organizacionales. Una sola organización puede ser separada en subdominios administrativos de menor tamaño. Cada subdominio recibe su zona. Todas las zonas en conjunto conforman al dominio en su totalidad.

Por ejemplo, `.ejemplo.org` es un dominio. Dentro de éste hay subdominios tales como `.ingenieria.ejemplo.org`, `.mercadotecnia.ejemplo.org`, `.ventas.ejemplo.org` y `.admin.ejemplo.org`. Cada uno de los cuatro subdominios tiene su propia zona. Y `.ejemplo.org` tiene algunos anfitriones dentro de sí que no quedan bajo ninguno de los subdominios; por lo tanto, tienen su propia zona. Como resultado, el dominio “`ejemplo.org`” está compuesto por un total de cinco zonas.

En el modelo más sencillo, donde un dominio no tiene subdominios, la definición de zona y dominio es la misma en función de la información acerca de anfitriones, configuraciones y demás.

El proceso para definir zonas en el archivo **named.conf** se trata en la siguiente sección.

CONFIGURACIÓN DE UN SERVIDOR DNS

En secciones anteriores aprendió acerca de las diferencias entre servidores primarios, secundarios y de caché. Recapitulando: los servidores de nombres primarios contienen las bases de datos con la información DNS más reciente para una zona. Cuando el administrador de zona quiere actualizar estas bases de datos, el servidor de nombre primario obtiene dicha actualización primero y el resto del mundo le pide a éste las actualizaciones. Los servidores secundarios en forma explícita llevan seguimiento de los primarios, y los primarios les notifican a los secundarios cuando ocurre un cambio. Se considera que los primarios y secundarios tienen la autoridad por igual en sus respuestas. Los servidores de caché no tienen registros autorizados, sólo tienen entradas de caché.

Definición de una zona primaria en el archivo named.conf

La sintaxis más básica para una entrada de configuración de zonas es como sigue:

```
zone nombre-del-dominio {
    type master;
    file ruta-y-nombre;
};
```

La **path-name** se refiere al archivo que contiene la información de la base de datos para la zona en cuestión. Por ejemplo, la creación de una zona para el dominio ejemplo.org, donde el archivo y la ubicación de la base de datos es `/var/named/ejemplo.org.db`, sería necesario crear la siguiente definición de zona en el archivo **named.conf**:

```
zone "ejemplo.org" {
    type master;
    file "ejemplo.org.db";
};
```

Note que la opción **directory** para el archivo **named.conf** en forma automática añadirá el prefijo correspondiente al nombre de archivo **ejemplo.org.db**. De manera que, si designa **directory /var/named**, el software del servidor buscará en forma automática la información para ejemplo.org en `/var/named/ejemplo.org.db`.

La definición de zona creada aquí es sólo una *referencia hacia delante*, esto es, el mecanismo por medio del cual otros pueden buscar un nombre a fin de obtener una dirección IP para un sistema debajo del dominio ejemplo.org que el servidor de nombres maneja. Son buenos modales de Red proporcionar también una representación IP-a-nOMBRE del anfitrión (también es necesario si quiere enviar correo electrónico a algunos sitios). Para hacer esto, debe proporcionar una entrada en el dominio in-addr.arpa.

El formato de una entrada in-addr.arpa consiste en los primeros tres octetos de su dirección IP, al revés, seguidos de in-addr.arpa. Suponiendo que la dirección IP de ejemplo.org empieza con 192.168.1, el dominio in-addr.arpa sería 1.168.192.in-addr.arpa. De este modo, el enunciado **zone** correspondiente dentro del archivo **named.conf** sería como sigue:

```
zone "1.168.192.in-addr.arpa" {
    type master;
    file "ejemplo.org.rev";
};
```

Note que los nombres de archivos (ejemplo.org.db y ejemplo.org.rev) utilizados aquí en las secciones de zona son completamente arbitrarios. Usted es totalmente libre de utilizar las reglas de denominación de archivos que quiera en tanto tengan sentido para usted.

La ubicación exacta de la sección de zona de nuestro ejemplo.org dentro del archivo **named.conf** será mostrada más adelante.

Opciones adicionales

Los dominios primarios también pueden utilizar algunas de las opciones de configuración del enunciado **options**. Dichas opciones son

- ▼ **check-names** (verifica nombres)
- **allow-update** (permite actualización)
- **allow-query** (permite solicitud)
- **allow-transfer** (permite transferencia)
- **notify** (notifica)
- ▲ **also-notify** (también notifica)

El uso de estas opciones en la configuración de una zona afectará solamente a esa zona.

Definición de una zona secundaria en el archivo named.conf

El formato de una entrada de zona para servidores secundarios es muy similar a aquella de servidores primarios. Para una resolución hacia delante, he aquí el formato:

```
zone nombre-del-dominio {  
    type slave;  
    masters lista-de-direcciones-IP; ;  
    file ruta-y-nombre;  
};
```

donde **domain-name** es el mismo nombre de zona especificado en el servidor primario, **IP-address-list** es la lista de direcciones IP donde está el servidor primario para esa zona, y **path-name** es la ruta completa hacia la ubicación donde el servidor mantendrá copias de los archivos de zona primarios.

Opciones adicionales

Una zona secundaria también puede utilizar algunas de las opciones del enunciado **options**. Éstas son:

- ▼ **check-names** (verifica nombres)
- **allow-update** (permite actualización)
- **allow-query** (permite solicitud)
- **allow-transfer** (permite transferencia)
- ▲ **max-transfer-time-in** (tiempo máximo de transferencia)

Definición de una zona de caché en el archivo named.conf

La configuración de caché es la más fácil de todas las configuraciones. También se necesita para toda configuración de un servidor DNS, aun cuando esté ejecutando un servidor primario o uno secundario. Ello es necesario con el fin de que el servidor pueda hacer una búsqueda recursiva en el árbol DNS cuando requiera encontrar otros anfitriones en Internet.

Para el caché de un servidor de nombres definimos tres secciones de zona. He aquí la primera entrada:

```
zone "." {  
    type hint;  
    file "root.hints";  
};
```

La primera entrada de la zona mostrada aquí es la definición de servidores de nombre raíz. La línea **type hint**; especifica que ésta es una entrada que define una zona de caché, y la línea **file "root.hints"**; especifica el archivo que preparará el caché con entradas que apunten a los servidores raíz. Siempre es posible obtener el último archivo root hints de <http://www.internic.net/zones/named.root>.

La segunda entrada de la zona de caché define la resolución de nombres para el anfitrión local y es como sigue:

```
zone "localhost" in {  
    type master;  
    file "localhost.db";  
};
```

La tercera entrada de la zona de caché define la búsqueda inversa para el anfitrión local. Ésta es la entrada inversa para resolver la dirección del anfitrión local (127.0.0.1) de vuelta en el nombre del anfitrión local.

```
zone "0.0.127.in-addr.arpa" {  
    type master;  
    file "127.0.0.rev";  
};
```

Poner estas entradas de zona en **/etc/named.conf** es suficiente para crear un servidor de caché DNS. Claro está que el contenido real de los archivos de la base de datos (**localhost.db**, **127.0.0.rev**, **ejemplo.org.db**, etc.) a los que se hace referencia en la directriz **file** también son de suma importancia. Las siguientes secciones examinarán la elaboración de la base de datos de manera más precisa.

Tipos de registros DNS

Esta sección abordará la elaboración de los archivos de la base de datos del servidor de nombres, esto es, los archivos que almacenan información específica pertinente a cada zona hospedada por el servidor. Los archivos de la base de datos consisten en su mayoría de los tipos de registro, por lo tanto, necesita comprender el significado y el uso de tipos de registro comunes para DNS tales como SOA, NS, A, PTR, CNAME, MX, TXT y RP.

SOA: Inicio de autoridad

El registro SOA (Start of Authority) inicia la descripción de las entradas DNS de un sitio. El formato de esta entrada es como sigue:

```
1     nombre.dominio. IN SOA sn.nombre.dominio. administrador.nombre.dominio. (
2           1999080801      ; número de serie
3           10800          ; tasa de actualización en segundos (3 horas)
4           1800           ; reintenta en segundos (30 minutos)
5           1209600         ; caduca en segundos (2 semanas)
6           604800          ; mínimo en segundos (1 semana)
7       )
```

NOTA Se añade la numeración de líneas al listado anterior para facilitar su lectura.

La primera línea contiene algunos detalles a los que debe prestar gran atención: **nombre.dominio.** se reemplaza por el nombre de su dominio. Por lo general éste es el mismo nombre que se especificó en la directriz **zone** en el archivo **/etc/named.conf**. Note ese último punto al final de **nombre.dominio.** Se supone que debe ir ahí —ciertamente, los archivos de configuración DNS son quisquillosos al respecto—. El punto final es necesario para que el servidor pueda diferenciar nombres de anfitrón relacionados de nombres de dominio completamente calificados (FQDN); por ejemplo, la diferencia entre servidorA y servidorA.ejemplo.org.

IN le dice al servidor de nombres que éste es un registro de Internet. Hay otros tipos de registro pero han pasado años sin que alguien los necesite. Puede ignorarlos con toda seguridad.

SOA le dice al servidor de nombres que éste es un registro de Inicio de autoridad.

El **sn.nombre.dominio** es el FQDN para el servidor de nombres de este dominio (es decir, el servidor donde este archivo residirá en última instancia). De nuevo, fíjese en el punto al final y no lo omita.

El **administrador.nombre.dominio** es la dirección de correo electrónico del administrador del dominio. Note la ausencia de @ en esta dirección. El símbolo @ es reemplazado por un punto. Así, la dirección referida en este ejemplo es administrador@nombre.dominio. El punto al final también se utiliza aquí.

El resto del registro empieza con el paréntesis que abre en la línea 1. La línea 2 es el número de serie. Se usa para decirle al servidor de nombres cuando ha sido actualizado el archivo. Tenga cuidado, olvidarse de incrementar este número cuando se hacen cambios es un error común en el proceso de administración de registros DNS (olvidarse de poner un punto en el lugar correcto es otro error común).

NOTA A fin de mantener los números de serie de modo que tengan sentido, utilice la fecha con el siguiente formato: YYYYMMDDxx. Las xx que están al final es un número adicional de dos dígitos que inician con 00, así que si usted hace actualizaciones múltiples en un solo día, puede aún decir cuál es cuál.

La línea 3 en la lista de valores es la tasa de actualización en segundos. Este valor le indica a los servidores DNS secundarios la frecuencia con la que deben preguntar al servidor primario si los registros se han actualizado.

La línea 4 es la tasa de reintento en segundos. Si el servidor secundario intenta pero no puede contactar al servidor DNS primario a fin de verificar la existencia de actualizaciones, el servidor secundario intenta de nuevo después del número de segundos especificado.

La línea 5 especifica la directriz de caducidad. Está pensada para servidores secundarios que tienen almacenados los datos de la zona. Les dice a estos servidores que, si no pueden contactar al servidor primario para una actualización, deben dimitir después de que transcurra el número de segundos especificado. De una a dos semanas es un buen valor para este intervalo.

El valor final (la línea 6, el mínimo) les dice a los servidores de caché cuánto deben esperar antes de que caduque una entrada en caso de que no puedan contactar al servidor DNS primario. De cinco a siete días es un buen parámetro para esta entrada.

SUGERENCIA No olvide poner el paréntesis que cierra (la línea 7) después del último valor.

NS: Servidor de nombres

El registro NS se usa para especificar cuáles servidores de nombres mantienen registros para esta zona. Si existen cualesquiera de los servidores secundarios a los que tiene planeado transferir zonas, éstos se deben especificar aquí. El formato de este registro es como sigue:

IN NS	sn1.nombre.dominio
IN NS	sn2.nombre.dominio

Puede tener tantos respaldos de servidores de nombre como desee para un dominio, cuando menos dos son una buena idea. La mayoría de los ISP están dispuestos a fungir como servidores DNS secundarios en caso de proveerle conectividad.

A: Registro de dirección

Éste es quizás el tipo de registro más común que puede encontrarse a campo abierto. El registro A se usa para proporcionar un mapeo de nombre de anfitrión hacia la dirección IP. El formato para el registro A es simple:

Nombre_anfitrión	IN A	Dirección_IP
------------------	------	--------------

Por ejemplo, un registro A para el anfitrión servidorB.ejemplo.org cuya dirección IP es 192.168.1.2 sería como sigue:

servidorB	IN A	192.168.1.2
-----------	------	-------------

Note que a cualquier nombre de anfitrión se le añade en forma automática el nombre del dominio listado en el registro SOA, a menos que este nombre de anfitrión termine con un punto. Retomando el ejemplo para el servidorB, si el registro SOA se define para ejemplo.org, entonces la referencia a servidorB será entendida como servidorB.ejemplo.org. Si definiera este campo como servidorB.ejemplo.org (sin el punto), el servidor de nombres entendería que se refiere a

servidorB.ejemplo.org.ejemplo.org., ¡lo cual quizás no sea lo que quería! Así que si requiere utilizar un FQDN, asegúrese de añadirle un punto.

PTR: Registro apuntador

El registro PTR (Pointer) es para llevar a cabo la resolución de nombres inversa, lo cual permite a alguien especificar una dirección IP y determinar el nombre del anfitrión correspondiente. El formato para este registro es muy similar al registro A, con la salvedad de que los valores están invertidos:

Dirección_IP IN PTR *Nombre_anfitrión*

La *Dirección_IP* puede tomar una de dos formas: sólo el último octeto de la dirección IP (dejando al servidor de nombres que en forma automática añada el sufijo con la información que tiene del nombre de dominio in-addr.arpa); o bien, la dirección IP completa añadiendo un punto. El *Nombre_anfitrión* debe llevar el FQDN completo. Por ejemplo, el registro PTR para el anfitrión servidorB sería como sigue:

192.168.1.2. IN PTR servidorB.ejemplo.org.

MX: Intercambio de correo

El registro MX (Mail Exchanger) se encarga de decirle a otros sitios acerca del servidor de correo de su zona. Si un anfitrión en su red genera un mensaje saliente con el nombre del anfitrión en éste, quien decida responder al mensaje no lo enviaría de regreso directamente a ese anfitrión. En vez de ello, el servidor que responde se asoma al registro MX para ese sitio y envía ahí el mensaje.

Cuando los sitios en Internet estaban primordialmente compuestos por sistemas basados en UNIX, con el Sendmail configurado como un anfitrión NULL (nulo) que reenviaba todo a un concentrador de correo, la falta de un registro MX no causaba ningún problema. Pero, conforme se unieron a la Red más sistemas que no son de UNIX, los registros MX se hicieron cruciales. Si un pc.nombre.dominio envía un mensaje utilizando un lector de correo basado en PC (que no acepte correo SMTP), es importante que la parte que contesta tenga una manera confiable de saber la identidad del servidor de correo de pc.nombre.dominio.

El formato del registro MX es como sigue:

nombredominio. IN MX *peso Nombre_anfitrión*

donde ***nombredominio.*** es el nombre del dominio del sitio (con un punto al final, claro); el ***peso*** es la importancia del servidor de correo (si existen varios servidores de correo, aquel con el número menor tiene precedencia sobre aquellos con números mayores); y el ***Nombre_anfitrión*** es, como puede intuir, el nombre del servidor de correo. Es importante que también el ***Nombre_anfitrión*** tenga el correspondiente registro A.

He aquí un ejemplo de esta entrada:

ejemplo.org. IN MX 10 smtp1
 IN MX 20 smtp2

Es usual que los registros MX aparezcan cerca del inicio de los archivos de configuración DNS. Si el nombre del dominio no se especifica, el nombre predeterminado se extrae del registro SOA.

CNAME: Nombre canónico

Los registros CNAME (Canonical Name) le permiten crear alias para los nombres de anfitriones. Un registro CNAME puede tomarse como un alias. Son útiles cuando quiere proveer un servicio altamente disponible con un nombre fácil de recordar pero quiere dar al anfitrión un nombre real.

Otro uso popular del registro CNAME es “crear” un nuevo servidor con un nombre fácil de recordar, sin necesidad de invertir en absoluto en un nuevo servidor. Un ejemplo: suponga que un sitio tiene un servidor Web con zabtsuj-content.ejemplo.org como su nombre de anfitrión. Puede argumentarse que zabtsuj-content.ejemplo.org no es un nombre fácil de recordar ni es amigable para el usuario. Así que, dado que el sistema es un servidor Web, un registro CNAME o alias de “www” puede crearse para el anfitrión. Esto sencillamente creará una representación del nombre no amigable de zabtsuj-content.ejemplo.org como un nombre más amable tal como www.ejemplo.org. Esto permitirá que todas las solicitudes que vayan a www.ejemplo.org sean transferidas de manera transparente al sistema que en realidad hospeda el contenido Web, esto es, zabtsuj-content.ejemplo.org.

He aquí el formato para el registro CNAME:

```
Nuevo_nombre_anfitrión IN CNAME viejo_nombre_anfitrión
```

Por ejemplo, retomando el escenario antes explicado, la entrada CNAME sería:

zabtsuj-content	IN	A	192.168.1.111
www	IN	CNAME	zabtsuj-content

RP y TXT: Las entradas de documentación

Algunas veces es útil proporcionar información de contacto como parte de su base de datos, no sólo como comentarios sino como registros reales que otros pueden consultar. Esto puede lograrse utilizando los registros RP y TXT.

Un registro TXT es una entrada de texto libre de formato en la cual puede colocar cualquier información que considere pertinente. Lo más usual es que sólo quiera poner información de contacto en estos registros. Cada registro TXT debe estar ligado a un nombre de anfitrión en particular. Por ejemplo,

```
servidorA.ejemplo.org. IN TXT "Contacto: El sabelotodo"  
IN TXT "SysAdmin/Androide"  
IN TXT "Voz: 999-999-9999"
```

El registro RP se creó como un contenedor explícito para la información de contacto de un anfitrión. Este registro enumera quién es la persona responsable para un anfitrión dado; he aquí un ejemplo:

```
servidorB.ejemplo.org. IN RP dirección-admin.ejemplo.org. ejemplo.org.
```

Así como son útiles estos registros, hoy en día son raros. Ello se debe a que se piensa que ofrecen mucha información acerca de un sitio que podría llevar a ataques basados en engaños o estafas conocidos como ingeniería social. Encontrará que tales registros pueden ser útiles para sus servidores DNS internos, pero debería dejarlos fuera del alcance de alguien que pudiera preguntarlos por Internet.

PREPARACIÓN DE LOS ARCHIVOS DE LA BASE DE DATOS BIND

Así que ahora ya sabe lo suficiente acerca de todos los tipos de registro DNS como para que inicie. Es momento de crear la base de datos real que alimentará al servidor.

El formato del archivo de la base de datos no es muy estricto que digamos, pero algunas reglas se han mantenido con el paso del tiempo. Apegarse a ellas le hará la vida más sencilla y allanará el camino al administrador que se haga cargo de su creación.

NOTA Siéntase libre de añadir comentarios. En este archivo, las líneas de comentarios empiezan con punto y coma. Aunque no hay mucho misterio acerca de lo que sucede en el archivo de la base de datos DNS, una historia de los cambios es una referencia útil sobre lo que trataba de lograr y por qué.

Los archivos de la base de datos son sus más importantes archivos de configuración. Es fácil crear las bases de datos de búsquedas hacia delante; lo que generalmente se queda fuera son las búsquedas inversas. Algunas herramientas como Sendmail y TCP-Wrappers harán búsquedas inversas sobre direcciones IP para ver de dónde vienen las personas. Es una cortesía común proporcionar esta información.

Cada archivo de base de datos debe empezar con una entrada \$TTL. Esta entrada le dice a BIND cuál es el valor para el tiempo de vida (time-to-live) de cada registro individual, cuando no se especifique en forma explícita (el TTL en el registro SOA es sólo para este registro). Después de la entrada \$TTL viene el registro SOA y cuando menos un registro NS. Todo lo demás es opcional (claro está, ¡“todo lo demás” es lo que hace útil al archivo!) Quizá le sea de ayuda el formato general siguiente:

```
$TTL  
registros SOA  
registros NS  
registros MX  
registros A y CNAME
```

Demos un recorrido por el proceso de construcción de un servidor DNS completo, de inicio a fin, para mostrar mejor cómo se reúne la información vista hasta ahora. Para este ejemplo, construiremos un servidor DNS para ejemplo.org que conseguirá los siguientes propósitos:

- ▼ Establecerá dos servidores de nombres: sn1.ejemplo.org y sn2.ejemplo.org.
- Actuará como un servidor esclavo para la zona ventas.ejemplo.org, donde el servidorB.ejemplo.org será el servidor maestro.
- Definirá registros A para servidorA, servidorB, smtp, sn1 y sn2.
- Definirá smtp.ejemplo.org como el intercambiador de correo (MX) para el dominio ejemplo.org.
- Definirá www.ejemplo.org como un nombre alternativo (CNAME) para servidorA.ejemplo.org y ftp.ejemplo.org para servidorB.ejemplo.org.
- ▲ Por último, definirá información de contacto para el servidorA.ejemplo.org.

Bien, Mr. Bond, ya tiene sus instrucciones. Obedezca las órdenes y cumpla con su misión. ¡Buena suerte!

División de los pasos individuales

Para lograr nuestro propósito y poner en funcionamiento un servidor DNS para ejemplo.org, necesitaremos llevar a cabo una serie de pasos. Trabajemos con ellos uno a la vez.

1. Asegúrese de que instaló el software del servidor DNS BIND como se describió anteriormente en este capítulo. Utilice el comando **rpm** para confirmar esto. Escriba:

```
[root@serverA ~]# rpm -q bind  
bind-9.*
```

NOTA Si construyó e instaló BIND desde el código fuente, entonces el comando **rpm** anterior no revelará nada porque la base de datos RPM no sabrá nada acerca de ello. Pero usted sí sabe... qué instaló y dónde. ¿O no?

2. Utilice cualquier editor de texto en el que se sienta cómodo para crear el archivo de configuración principal del servidor DNS, esto es, el archivo **/etc/named.conf**. Introduzca el siguiente texto en el archivo:

```
options {  
    directory      "/var/named";  
    dump-file     "/var/named/data/cache_dump.db";  
    statistics-file  "/var/named/data/named_stats.txt";  
    notify        yes;  
};  
  
# Las siguientes definiciones de zona no necesitan ninguna modificación. La primera  
# es la definición de los servidores de nombres raíz y configura nuestro servidor  
# como un servidor DNS con capacidad de caché.  
# La segunda define el anfitrión local.  
# La tercera zona define la búsqueda inversa para el anfitrión local.  
zone "." in {  
    type hint;  
    file "root.hints";  
};  
zone "localhost" in {  
    type master;  
    file    "localhost.db";  
};  
zone "0.0.127.in-addr.arpa" in {  
    type master;  
    file    "127.0.0.rev";  
};  
# La definición de zona que sigue es para el dominio en el cual nuestro servidor  
# tiene autoridad, esto es, el nombre de dominio ejemplo.org.  
zone "ejemplo.org" {  
    type master;  
    file    "ejemplo.org.db";  
};  
  
# Enseguida definimos la zona para el dominio in-addr.arpa, para el sitio ejemplo.org.
```

```
zone "1.168.192.in-addr.arpa" {
    type master;
    file "ejemplo.org.rev";
};

# Enseguida está la entrada del subdominio para el cual este servidor es un
# servidor esclavo. La dirección IP del servidor maestro ventas.ejemplo.org es
# 192.168.1.2

zone "ventas.ejemplo.org" {
    type slave;
    file "ventas.ejemplo.org.bk";
    masters {192.168.1.2};
};

};
```

3. Guarde el archivo anterior como **/etc/named.conf** y salga del editor de textos.
 4. Enseguida, necesitaremos crear los archivos de la base de datos a los que se hizo referencia en el archivo **/etc/named.conf**. En particular, los archivos que crearemos son **root.hints**, **localhost.db**, **127.0.0.rev**, **ejemplo.org.db** y **ejemplo.org.rev**. Todos los archivos serán almacenados en el directorio de trabajo de BIND, **/var/named/**. Los iremos creando conforme aparecen desde el inicio del archivo **named.conf** hasta el final.
 5. Afortunadamente, no tendremos que crear el archivo root hints en forma manual. Descargue la copia más reciente del archivo root hints desde Internet. Utilice el comando **wget** para descargarlo y copiarlo en el directorio apropiado. Escriba

```
[root@serverA ~]# wget -O /var/named/root.hints \
http://www.internic.net/zones/named.root
```

6. Utilice cualquier editor de textos en el que se sienta cómodo para crear el archivo de zona para el anfitrión local. Éste es el archivo **localhost.db**. Introduzca el texto que sigue en el archivo:

```
$TTL 1W
@           IN SOA  localhost  root (
                           2006123100      ; serie
                           3H            ; actualizaciones (3 horas)
                           30M           ; reintentos (30 minutos)
                           2W            ; caduca (2 semanas)
                           1W )          ; mínimo (1 semana)
IN NS        @
IN A         127.0.0.1
```

7. Guarde el archivo anterior como **/var/named/localhost.db** y salga del editor de texto.
 8. Utilice cualquier editor de textos a fin de crear el archivo de zona para la zona de búsqueda inversa del anfitrión local. Éste es el archivo **127.0.0.rev**. Introduzca el texto que sigue en el archivo:

```
$TTL 1W
@           IN SOA      localhost.  root.localhost.  (
                        2006123100 ; serie
                        3H          ; actualizaciones
                        30M         ; reintentos
```

```

          2W           ; caduca
          1W )         ; mínimo

1           IN NS      localhost.
           IN PTR     localhost.

```

SUGERENCIA Es posible utilizar valores de tiempo abreviados en BIND. Por ejemplo, 3H significa 3 horas, 2W quiere decir 2 semanas, 30M implica 30 minutos, etcétera.

9. Guarde el archivo anterior como **/var/named/127.0.0.rev** y salga del editor de texto.
10. Enseguida, prepare el archivo de la base de datos para la zona principal en cuestión, esto es, el dominio **ejemplo.org**. Utilice un editor de texto para crear el archivo **ejemplo.org.db** e introduzca el texto que sigue en el archivo:

```

$TTL 1W
@           IN SOA ns1.ejemplo.org.  root (
                                         2006123100      ; serie
                                         3H              ; actualizaciones (3 horas)
                                         30M             ; reintentos (30 minutos)
                                         2W              ; caduca (2 semanas)
                                         1W )            ; mínimo (1 semana)

           IN   NS      ns1.ejemplo.org.
           IN   NS      ns2.ejemplo.org.
           IN   MX 10    smtp.ejemplo.org.

ns1          IN   A       192.168.1.1  ; servidor de nombres primario
ns2          IN   A       192.168.1.2  ; servidor de nombres secundario
servidorA    IN   A       192.168.1.1
servidorB    IN   A       192.168.1.2
smtp          IN   A       192.168.1.25 ; servidor de correo
www           IN   CNAME  servidorA    ; servidor web
ftp            IN   CNAME  servidorB    ; servidor ftp
servidorA    IN   TXT    "Fax: 999-999-9999"

```

11. Guarde el archivo anterior como **/var/named/ejemplo.org.db** y salga del editor de texto.
12. Después, elabore el archivo de zona para la búsqueda inversa en la zona **ejemplo.org**. Utilice un editor de texto para crear el archivo **ejemplo.org.rev** e introduzca el siguiente texto:

```

$TTL 1W
@           IN SOA ns1.ejemplo.org.  root (
                                         2006123100      ; serie
                                         3H              ; actualizaciones (3 horas)
                                         30M             ; reintentos (30 minutos)
                                         2W              ; caduca (2 semanas)
                                         1W )            ; mínimo (1 semana)

           IN   NS      ns1.ejemplo.org.
           IN   NS      ns2.ejemplo.org.

1           IN   PTR     servidorA.ejemplo.org. ; información inversa para servidorA
2           IN   PTR     servidorB.ejemplo.org. ; información inversa para servidorB
25          IN   PTR     smtp.ejemplo.org.      ; inversa para servidor de correo

```

13. No tiene que crear ningún archivo para configurar el servidor como el secundario de `ventas.ejemplo.org`. Sólo necesita añadir las entradas que ya tiene en el archivo `named.conf` (aunque los archivos de registro se quejarán porque no podrán contactar al servidor maestro, esto está bien; ello es así porque sólo ha puesto en funcionamiento un maestro primario para la zona en la cual este mismo servidor tiene la autoridad).

El siguiente paso mostrará cómo iniciar el servicio `named`. Pero, como el software BIND es muy escrupuloso con los puntos y los puntos y comas, y porque quizás haya tenido que introducir manualmente todos los archivos de configuración, las posibilidades de que haya cometido errores de dedo son invariablemente altas (o quizás nosotros las cometimos ☺). Así que su mejor apuesta será monitorear el sistema de archivos de registro para ver los mensajes de error conforme éstos se generan en tiempo real.

14. Utilice el comando `tail` en otra ventana de terminal para ver los registros y luego ejecute el comando del paso 15 en una ventana separada de manera que pueda ver ambas simultáneamente. En una de las nuevas ventanas de terminal escriba

```
[root@serverA named]# tail -f /var/log/messages
```

15. Estamos listos para iniciar el servicio `named`. Utilice el comando `service` para iniciar el servicio. Escriba

```
[root@serverA named]# service named start  
Starting named: [ OK ]
```

SUGERENCIA En SuSE Linux, el comando equivalente sería

```
[root@serverA] # rcnamed start
```

16. Si obtiene un montón de errores en los registros del sistema, encontrará que éstos por lo general le dicen el número de línea y(o) el tipo de error. Así que arreglar los errores no debe ser demasiado difícil. Sólo regrese al editor de texto y añada los puntos y puntos y comas donde hagan falta. Otro error común es equivocarse al escribir las directrices de los archivos de configuración, esto es, escribir `master` en vez de `masters`; aunque ambas son directrices válidas, cada una se utiliza en contextos diferentes.

SUGERENCIA Si ha cambiado los archivos de configuración de BIND (ya sea el principal `named.conf` o los archivos de la base de datos), debe ordenarle al programa que vuelva a leerlos enviando la señal HUP al proceso `named`. Empiece por encontrar el proceso ID para el proceso o `named`. Esto se puede hacer al buscarlo en `/var/run/named/named.pid`. Si no lo ve en la ubicación usual, puede ejecutar el siguiente comando para obtenerlo.

```
[root@serverA ~]# ps -C named  
PID TTY TIME CMD  
7706 ? 00:00:00 named
```

El valor debajo de la columna PID (Process ID) es el identificador del proceso `named`. Éste es el PID al cual debemos enviar la señal HUP. Ahora puede hacerlo escribiendo

```
[root@serverA ~]# kill -HUP 7706
```

Por supuesto, reemplazar **7706** con el ID proceso de su información.

17. Por último, quizá quiera asegurarse de que el servicio de su servidor DNS empiece durante el siguiente reinicio del sistema. Utilice el comando **chkconfig**. Escriba

```
[root@serverA named]# chkconfig named on
```

La siguiente sección lo llevará por el uso de las herramientas que pueden ser utilizadas para probar/interrogar un servidor DNS.

LA CAJA DE HERRAMIENTAS DNS

Esta sección describe unas cuantas herramientas con las que querrá familiarizarse conforme trabaja con DNS. Le ayudarán a fulminar problemas con celeridad.

host

La herramienta **host** es en realidad una muy fácil de utilizar. Su funcionalidad puede extenderse al utilizarse con varias opciones.

Enseguida se muestran sus opciones y sintaxis:

```
host [-aCdldrTwv] [-c class] [-n] [-N ndots] [-t type] [-W time]
      [-R number] hostname [server]
-a es equivalente a -v -t *
-c especifica el tipo de solicitud para datos non-IN data
-C compara registros SOA en servidores de nombre con autoridad
-d es equivalente a -v
-l lista todos los anfitriones en un dominio, utilizando AXFR
-i Utiliza la forma vieja IN6.INT de la búsqueda inversa de IPv6
-N cambia el número de puntos permitido antes de que la búsqueda raíz termine
-r inhabilita el procesamiento recursivo
-R especifica el número de reintentos para paquetes UDP
-t especifica el tipo de solicitud
-T habilita el modo TCP/IP
-v habilita la salida de texto explicativo
-w especifica una espera eterna por una respuesta
-W especifica cuánto esperar por una respuesta
```

En su forma más simple, **host** permite representar nombres de anfitriones en direcciones IP desde la línea de comandos. Por ejemplo,

```
[root@serverA named]# host internic.net
internic.net has address 198.41.0.6
```

También podemos utilizar **host** para realizar búsquedas inversas. Por ejemplo,

```
[root@serverA named]# host 198.41.0.6
6.0.41.198.in-addr.arpa domain name pointer rs.internic.net.
```

dig

El rastreador de información sobre dominios, **dig**, es una gran herramienta acerca de los servidores DNS. Es la herramienta que tiene la bendición y el sello oficial del grupo BIND.

Enseguida se muestran su sintaxis y algunas de sus opciones:

```
dig [@global-server] [domain] [q-type] [q-class] {q-opt}
    {global-d-opt} host [@local-server] {local-d-opt}
    [ host [@local-server] {local-d-opt} [...] ]
```

Donde: domain está en el Domain Name System

q-class	es uno de (in,hs,ch,...) [default: in]
q-type	es uno de (a,any,mx,ns,soa,hinfo,axfr,txt,...) [default:a]
q-opt	es uno de:
-x dot-notation	(atajo para búsquedas in-addr)
-i	(IP6.INT IPv6 búsquedas inversas)
-f filename	(modo de trabajo por lotes)
-b address	(liga hacia la dirección fuente)
-p port	(especifica el número de puerto)
-t type	(especifica el tipo de solicitud)
-c class	(especifica el tipo de solicitud)
-k keyfile	(especifica la clase de solicitud)
-y name:key	(especifica la llave named tsig base 64)
d-opt	es una de la forma +palabra clave[=valor], donde palabra clave es:
+[no]vc	(Modo TCP)
+[no]tcp	(Modo TCP, sintaxis alterna)
+time=###	(Ajusta el tiempo límite para la solicitud) [5]
+tries=###	(Ajusta el número de intentos UDP) [3]
+domain=###	(Ajusta el nombre de dominio predeterminado)
+[no]recurse	(Modo recursivo)
+[no]ignore	(No revierta a TCP para respuestas TC.)
+[no]fail	(No intente con el siguiente servidor en caso de falla, SERVFAIL)
+[no]besteffort	(Intente analizar inclusive mensajes ilegales)
+[no]aaonly	(Active bandera AA en solicitud)
+[no]cmd	(Controle visualización de la línea de comandos)
+[no]comments	(Controle visualización de líneas de comentarios)
+[no]question	(Controle visualización de la pregunta)
+[no]answer	(Controle visualización de la respuesta)
+[no]authority	(Controle visualización de la autoridad)
+[no]additional	(Controle visualización de la adicional)
+[no]stats	(Controle visualización de estadísticas)

+ [no]short	(Inhabilite todo excepto forma corta de respuesta)
+ [no]all	(Active o libere todas las banderas de visualización)
+ [no]qr	(Imprima pregunta antes de enviarla)
+ [no]nssearch	(Busque todos los servidores de nombre autorregulados)
+ [no]identify	(Identificación de respondedores en respuestas cortas)
+ [no]trace	(Rastree delegación desde la raíz)
+ [no]dnssec	(Solicite registros DNSSEC)
+ [no]multiline	(Imprima registros en formato expandido)

global d-opts y servers (antes del nombre del anfitrión) afectan todas las solicitudes local d-opts y servers (después del nombre del anfitrión) afectan sólo esa búsqueda.

En resumen, el uso de **dig** es

dig @servidor dominio tipo-de-solicitud

donde **@servidor** es el nombre del servidor DNS al que quiere interrogar, **dominio** es el nombre del dominio por el cual usted quiere preguntar, y **tipo-de-solicitud** es el tipo de registro que está intentando obtener (A, MX, NS, SOA, HINFO, TXT, ANY, etc.).

Por ejemplo, a fin de obtener el registro MX para el dominio ejemplo.org establecido anteriormente en el proyecto de servidor DNS que pusimos en funcionamiento, escribiría el comando **dig** como sigue:

```
[root@serverA ~]# dig @localhost ejemplo.org MX
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
;; QUESTION SECTION:
;ejemplo.org.           IN      MX
;; ANSWER SECTION:
ejemplo.org.          604800  IN      MX      10 smtp.ejemplo.org.
;; AUTHORITY SECTION:
ejemplo.org.          604800  IN      NS      ns1.ejemplo.org.
ejemplo.org.          604800  IN      NS      ns2.ejemplo.org.
;; ADDITIONAL SECTION:
smtp.ejemplo.org.    604800  IN      A       192.168.1.25
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
```

Para interrogar nuestro servidor DNS local por los registros A para el dominio yahoo.com, escriba

```
[root@serverA ~]# dig @localhost yahoo.com
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
;; QUESTION SECTION:
;yahoo.com.             IN      A
;; ANSWER SECTION:
yahoo.com.            300     IN      A      216.109.112.135
yahoo.com.            300     IN      A      66.94.234.13
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
```

NOTA Se dará cuenta de que, para el comando precedente, no se especificó el tipo de solicitud, esto es, no solicitamos en forma explícita el registro tipo A. El comportamiento predeterminado de **dig** es suponer que el solicitante quiere un registro tipo A cuando no se especifica nada. También se dará cuenta de que estamos solicitando a nuestro servidor DNS información del dominio yahoo.com. Nuestro servidor, claro está, no tiene la autoridad para el dominio yahoo.com pero, como también lo configuramos como un servidor DNS con capacidad de caché, es capaz de obtener la respuesta apropiada para nosotros desde los servidores DNS apropiados.

Para relanzar el comando previo pero suprimiendo la verborrea mediante una de las opciones de **dig** (+short), escriba

```
[root@serverA ~]# dig +short @localhost yahoo.com
66.94.234.13
216.109.112.135
```

Para preguntar al servidor de nombres local por la información de búsqueda inversa (PTR RR) para 192.168.1.1, escriba

```
[root@serverA ~]# dig -x 192.168.1.1 @localhost
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
;; QUESTION SECTION:
;1.1.168.192.in-addr.arpa.      IN      PTR
;; ANSWER SECTION:
1.1.168.192.in-addr.arpa. 604800 IN      PTR      servidorA.ejemplo.org.
;; AUTHORITY SECTION:
1.168.192.in-addr.arpa. 604800 IN      NS       ns1.ejemplo.org.
1.168.192.in-addr.arpa. 604800 IN      NS       ns2.ejemplo.org.
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
```

El programa **dig** es increíblemente poderoso. Sus opciones son demasiado numerosas como para tratarlas aquí de manera adecuada. Debería leer la página “man” que se instaló con **dig** para aprender a utilizar algunas de sus características más avanzadas.

nslookup

La utilidad **nslookup** es una de las que existen en varias plataformas de sistemas operativos. Y quizás sea una de las que la mayoría de las personas conoce. Su uso es muy sencillo. Puede ser utilizada en forma interactiva y en forma directa, es decir, desde la línea de comandos.

El modo interactivo se activa cuando el comando no recibe ningún argumento. Escribir **nslookup** por sí sólo en la línea de comandos le llevará al shell **nslookup**. Para salir del modo interactivo, escriba **exit** en el mensaje de comandos.

SUGERENCIA Cuando **nslookup** se usa en modo interactivo, el comando para salir de la herramienta es **exit**. Pero, de manera instintiva, la gente acostumbra utilizar el comando **quit** para intentar salir del modo interactivo. **nslookup** pensará que está pidiendo información DNS de un anfitrión llamado “quit”. Después de un rato, terminará la búsqueda. Puede crear un registro que le recordará de inmediato al usuario cuál es el comando apropiado para salir. Una entrada como la siguiente en el archivo de zona será suficiente:

```
utilice-exit-para-salir      IN A          127.0.0.1  
quit                         IN CNAME      utilice-exit-para-salir
```

Con la entrada precedente en el archivo de zona, siempre que alguien pregunte a su servidor DNS utilizando el modo interactivo de **nslookup** y por error escriba el comando **quit** para salir, el usuario recibirá un amable recordatorio que le dice “*utilice-exit-para-salir*”.

En el modo directo, el uso del comando se resume enseguida:

```
nslookup [ -option ] [ name | - ] [ server ]
```

Por ejemplo, para utilizar **nslookup** en modo directo y preguntar a nuestro servidor local por información acerca del anfitrión `www.ejemplo.org`, escriba

```
[root@serverA ~]# nslookup www.ejemplo.org localhost  
Server:      localhost  
Address:     127.0.0.1#53  
  
www.ejemplo.org canonical name = servidorA.ejemplo.org.  
Name:        servidorA.ejemplo.org  
Address:    192.168.1.1
```

NOTA El grupo desarrollador de BIND frunce la ceja sobre el uso de la herramienta **nslookup**. Oficialmente es despreciada.

whois

Este comando se usa para determinar la propiedad de un dominio. La información acerca del propietario de un dominio no es una parte obligatoria de sus registros ni se acostumbra colocarla en los registros TXT o RP. Así que necesitará reunir esta información utilizando la técnica **whois**, misma que reporta el propietario real del dominio, su dirección de correo-tortuga, dirección de correo electrónico, y los números telefónicos del contacto técnico.

Demos un recorrido por un ejemplo con el cual se obtiene información acerca del dominio `yahoo.com`. Escriba

```
[root@serverA ~]# whois yahoo.com  
[Querying whois.internic.net]  
[Redirected to whois.alldomains.com]
```

```
[Querying whois.alldomains.com]
[whois.alldomains.com]
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
Registrant:
    Yahoo! Inc.
    (DOM-272993)
Technical Contact, Zone Contact:
    Domain Administrator
    (NIC-1372925)
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
Created on.....: 1995-Jan-18.
Expires on.....: 2012-Jan-19.
Record last updated on...: 2005-Apr-05 16:34:22.
Domain servers in listed order:
NS4.YAHOO.COM          63.250.206.138
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
```

nsupdate

Ésta es una utilidad DNS poderosa que por lo general se olvida. Se usa para presentar solicitudes de actualización de Dynamic DNS (DDNS) (DNS Dinámico) a un servidor DNS. Permite que los registros de recursos (RR) se añadan o eliminen de una zona sin editar a mano los archivos de la base de datos de esa zona.

Esto es especialmente útil porque las zonas tipo DDNS no deberían ser editadas o actualizadas a mano, ya que los cambios manuales son propensos a entrar en conflicto con las actualizaciones dinámicas que se mantienen en forma automática en los archivos del diario, lo cual puede dar por resultado que los datos de la zona se corrompan.

El programa **nsupdate** lee entradas de un archivo con un formato especial o mediante entradas estándar. La sintaxis del comando es

```
nsupdate [ -d ] [[ -y keyname:secret ] [ -k keyfile ] ] [-v] [filename ]
```

La herramienta rndc

Esta utilidad es el famoso remote name daemon (rndc) (control demonio de control de nombre remoto). Es muy útil para controlar el servidor de nombres y también para depurar problemas con dicho servidor.

El programa **rndc** se puede usar para administrar en forma segura el servidor de nombres. Para hacer esto es preciso crear un archivo de configuración independiente debido a que toda la comunicación con el servidor es autenticada con firmas digitales que dependen de un secreto compartido y éste, por lo general, se almacena en un archivo de configuración y se llama **/etc/rndc.conf**. Será necesario crear ese secreto compartido entre la herramienta y el servidor de nombres mediante herramientas como **rndc-confgen** (no abordamos esta característica aquí).

Se resume enseguida el uso de **rndc**:

```
rndc [-c config] [-s server] [-p port]
      [-k key-file] [-y key] [-V] command
command es uno de los siguientes:
```

```
reload      Vuelve a cargar el archivo de configuración y las zonas.  
reload zone [class [view]]  
           Vuelve a cargar una sola zona.  
refresh zone [class [view]]  
           Agenda de mantenimiento inmediato para la zona.  
reconfig    Vuelve a cargar el archivo de configuración y sólo las nuevas  
           zonas.  
stats       Guarda estadísticas del servidor en el archivo de estadísticas.  
querylog   Conmuta el registro de solicitudes.  
dumpdb     Vacía caché(s) al archivo de vaciado (named_dump.db).  
stop        Guarda actualizaciones pendientes a archivos maestros y detiene  
           el servidor.  
halt        Detiene el servidor sin guardar actualizaciones pendientes.  
trace       Incrementa en uno el nivel de depuración.  
trace level Cambia el nivel de depuración.  
notrace    Ajusta el nivel de depuración a 0.  
flush      Limpia todas las memorias de caché del servidor.  
flush [view] Limpia las memorias de caché del servidor para una vista.  
status     Muestra el estado del servidor.
```

Por ejemplo, puede utilizar **rndc** para ver el estado del servidor DNS. Escriba

```
[root@serverA ~]# rndc status  
number of zones: 7  
debug level: 0  
xfers running: 0  
xfers deferred: 0  
soa queries in progress: 1  
query logging is OFF  
server is up and running
```

Si, por ejemplo, hace cambios en el archivo de la base de datos de la zona (**/var/names/example.org.db**) para una de las zonas bajo su control (esto es, **ejemplo.org**) y quiere volver a cargar sólo esa zona sin reiniciar todo el servidor DNS, puede emitir el comando **rndc** con la opción mostrada enseguida:

```
[root@serverA ~]# rndc reload example.org
```

NOTA ¡Recuerde hacer incrementar el número de serie de la zona después de hacer cambios en ésta!

CONFIGURACIÓN DE CLIENTES DNS

En esta sección ¡nos adentraremos en el desenfrenado y emocionante proceso de configurar clientes DNS! Quizá exageramos; no son tan emocionantes, pero nadie niega la importancia que tienen para la infraestructura de cualquier sitio en red.

El Resolver

Hasta ahora hemos estudiado los servidores y el árbol DNS como un todo. La otra parte de esta ecuación es, claro, el cliente, el anfitrión que está contactando al servidor DNS para obtener una representación de un nombre de anfitrión en una dirección IP.

NOTA Quizá haya notado antes que, en la sección “La caja de herramientas DNS”, las solicitudes que hicimos, en su mayoría, se hicieron hacia un servidor DNS llamado “localhost”. Este servidor, claro está, es el sistema local cuyo shell ejecutan los comandos solicitados. En nuestro caso, ¡ojalá que este sistema fuera servidorA.ejemplo.org! La razón por la cual especificamos con precisión el servidor DNS para ser utilizado es que, de manera predeterminada, el sistema pregunta sea cual sea el servidor DNS definido en el anfitrión. Y si sucede que el servidor DNS de su anfitrión es algún servidor DNS aleatorio que su ISP le ha asignado, algunas de las solicitudes fallarán porque el servidor DNS de su ISP no sabrá acerca de la zona que usted administra y controla de manera local. Así que, si configuramos nuestro servidor DNS local para procesar todo tipo de solicitudes tipo DNS, entonces ya no tendremos que especificar manualmente “localhost”. A esto se le llama configurar el *Resolver*.

En Linux, el *Resolver* maneja el lado del cliente del DNS. Éste es en realidad parte de una biblioteca de funciones de programación en C que se *liga* a un programa cuando éste inicia. Debido a que todo esto ocurre en forma automática y transparente, el usuario no tiene que saber nada al respecto. Es sencillamente un poco de magia que permite iniciar la navegación en Internet.

Desde la perspectiva del administrador, la configuración del cliente DNS no es magia, sino muy directa. Solamente hay dos archivos involucrados: `/etc/resolv.conf` y `/etc/nsswitch.conf`.

El archivo `/etc/resolv.conf`

Este archivo contiene la información necesaria para que el cliente sepa cuál es su servidor DNS local (todo sitio debe tener, cuando menos, su propio servidor DNS de caché). Este archivo consta de dos líneas. La primera indica el dominio de búsqueda predeterminado, y la segunda indica la dirección IP del servidor de nombres del anfitrión.

El *dominio de búsqueda predeterminado* se aplica para la mayoría de los sitios que tienen sus propios servidores locales. Cuando se especifica este dominio de modo predeterminado, el lado del cliente anexará en forma automática el nombre de este dominio al sitio solicitado y allí comprobará primero. Por ejemplo, si especifica que su dominio predeterminado es yahoo.com y luego intenta conectarse al nombre del anfitrión “my”, el software del cliente intentará contactar en forma automática my.yahoo.com. Utilizando el mismo valor predeterminado, si trata de contactar al anfitrión, www.stat.net, el software intentará www.stat.net.yahoo.com (un nombre de anfitrión perfectamente legal), encontrará que no existe, e intentará con www.stat.net por sí sólo (el cual sí existe).

Desde luego, puede proporcionar varios dominios predeterminados. Sin embargo, hacerlo retardará un poco el proceso de solicitud, pues cada dominio necesitará ser revisado. Por ejemplo, si ambos ejemplo.org y stanford.edu quedan especificados, y lleva a cabo una solicitud en www.stat.net, obtendrá tres solicitudes: www.stat.net.yahoo.com, www.stat.net.stanford.edu y www.stat.net.

El formato del archivo `/etc/resolv.conf` es el siguiente:

```
search domainname  
nameserver IP-address
```

donde `domainname` es el nombre del dominio predeterminado que se buscará, e `IP-address` es la dirección IP de su servidor DNS. Por ejemplo, éste es un archivo `/etc/resolv.conf` muestra:

```
search example.org  
nameserver 127.0.0.1
```

Así, cuando se requiere la solicitud de búsqueda de un nombre para `servidorB.example.org`, sólo se necesita la parte del anfitrión, esto es, `servidorB`. El sufijo `example.org` se añadirá en forma automática a la solicitud. Desde luego, ello es válido en su sitio local, ¡donde tiene control sobre la forma como se configura a los clientes!

El archivo `/etc/nsswitch.conf`

Este archivo le dice al sistema dónde debe buscar cierta clase de información de configuración (*servicios*). Cuando se definen varias ubicaciones, el archivo `/etc/nsswitch.conf` también especifica el orden en el cual la información se puede encontrar mejor. Los archivos que típicamente se configuran para utilizar el archivo `/etc/nsswitch.conf` incluyen el archivo de contraseña, el de grupo y el de anfitriones (para ver una lista completa, abra el archivo en su editor de texto favorito).

El formato del archivo `/etc/nsswitch.conf` es sencillo. El nombre del servicio aparece primero en una línea (note que `/etc/nsswitch.conf` sirve para más cosas que sólo las búsquedas de nombre de anfitriones), seguido de dos puntos. Enseguida vienen las ubicaciones que contienen información. Si se identifican varias ubicaciones, las entradas se listan en el orden en el que el sistema necesita llevar a cabo la búsqueda. Entradas válidas para las ubicaciones son `files`, `nis`, `dns`, `[NOTFOUND]` y `NISPLUS`. Los comentarios empiezan con un signo de número (#).

Por ejemplo, si abre un archivo con su editor favorito, verá una línea similar a esta:

```
hosts:      files nisplus nis dns
```

Esta línea le dice al sistema que todas las búsquedas de nombres de anfitrión empiezan primero con el archivo `/etc/hosts`. Si la entrada no se puede encontrar ahí, se verifica NISPLUS. Si no puede encontrar el anfitrión vía NISPLUS, el NIS regular se revisa, y así hasta agotar las opciones definidas. Es posible que NISPLUS no se esté ejecutando en su sitio y quiera que el sistema verifique registros DNS antes de que revise registros NIS. En este caso, la línea cambiaría a

```
hosts:      files dns nis
```

Y eso es todo. Guarde su archivo y el sistema detectará los cambios en forma automática.

La única recomendación para esta línea es que la opción para el archivo de anfitriones (`files`) siempre debe aparecer primero en el orden de búsqueda.

¿Cuál es el orden preferido para NIS y DNS? Ello dependerá en buena medida de las características particulares del sitio. Si debe buscar el nombre del anfitrión con DNS antes de intentar NIS dependerá de si el servidor DNS está más cerca que el servidor NIS en función de la conectividad de red, si un servidor es más rápido que el otro, consideraciones sobre firewalls, sobre políticas de sitio y otros factores similares.

El uso de [NOTFOUND=action]

En el archivo `/etc/nsswitch.conf` verá entradas que terminan en `[NOTFOUND=action]`. Ésta es una directriz especial que le permite detener el proceso de la búsqueda de información una vez que el sistema ha revisado todas las entradas previas. La acción puede ser `return` (regresar) o `continue` (continuar). La acción predeterminada es continuar.

Por ejemplo, si su archivo contiene la línea `hosts: files [NOTFOUND=return] dns nis`, el sistema intentará buscar la información del anfitrión sólo en el archivo `/etc/hosts`. Si la información requerida no se encuentra ahí, DNS y NIS no se revisarán.

La configuración del cliente

Vayamos paso a paso por el proceso de configuración de un cliente Linux que utilizará un servidor DNS. Suponemos que se está usando un servidor DNS en el servidorA y que estamos configurando al mismo servidorA como cliente. De entrada, esto puede sonar raro pero es importante recalcar que sólo porque el sistema ejecuta el servidor DNS ello no significa que no pueda ejecutar el cliente. Piense en el caso de un servidor Web: ¡sólo porque el sistema ejecuta Apache no significa que no pueda ejecutar Firefox en la misma máquina y acceder 127.0.0.1!

Al dividir la tarea de configuración del cliente en algunos pasos, vemos lo siguiente:

1. Edite `/etc/resolv.conf` y ajuste la entrada `nameserver` de manera que apunte a su servidor DNS. Para nuestro ejemplo:

```
search example.org  
nameserver 127.0.0.1
```

2. Busque en todo el archivo `/etc/nsswitch.conf` para asegurarse de que se consulta el DNS para la resolución de nombres de anfitrión. Edite `etc/nsswitch.conf` de manera que haga búsquedas de nombres.

```
[root@serverA ~]# grep "hosts" /etc/nsswitch.conf  
hosts: files dns
```

Si no tiene listado `dns` como en este resultado, use cualquier editor de textos para incluirlo en la línea de anfitriones.

3. Pruebe la configuración con la herramienta `dig`. Escriba

```
[root@serverA ~]# dig +short servidorA.ejemplo.org  
192.168.1.1
```

Note que no tuvo que especificar en forma explícita el nombre del servidor que se va a usar (como `@localhost`) en la solicitud anterior.

RESUMEN

En este capítulo cubrimos toda la información necesaria para poner en funcionamiento varios tipos de servidores DNS. Analizamos

- ▼ La resolución de nombres sobre Internet.
- Cómo obtener e instalar el servidor de nombres BIND.
- El archivo **/etc/hosts**.
- El proceso de configuración de un cliente Linux para que pueda utilizar DNS.
- La configuración de servidores DNS para que funcionen como servidores primarios, secundarios y de caché.
- Varios tipos de registros DNS.
- Opciones de configuración en el archivo **named.conf**.
- Herramientas que se utilizan en conjunto con el servidor DNS para localización de problemas.
- ▲ Fuentes adicionales de información.

Con la información disponible en la documentación BIND sobre cómo el servidor se debe configurar, además de los archivos de configuración reales para un servidor completo que fueron presentados en este capítulo, usted debe ser capaz de salir y llevar a cabo una instalación completa, de inicio a fin.

Como cualquier software, nada es perfecto y algunos problemas pueden surgir con BIND y los archivos y programas relacionados tratados aquí. Recuerde que para obtener información adicional debe visitar el sitio Web principal del BIND (<http://www.isc.org>), además de otros listados de correo dedicados a DNS y el software BIND.

CAPÍTULO 17



FTP

El File Transfer Protocol (FTP) (Protocolo de transferencia de archivos) existe en Internet desde 1971, aproximadamente. Es notable que desde entonces este protocolo casi no ha sufrido cambios. Por otro lado, los clientes y los servidores se han mejorado y refinado con regularidad casi permanente. Este capítulo cubre un paquete de software llamado vsftpd, Very Safe FTP Daemon (Demonio FTP Bastante Seguro).

El programa vsftpd es un servidor FTP muy popular que está siendo utilizado por sitios FTP importantes tales como kernel.org, redhat.com, isc.org y openbsd.org. El hecho de que estos sitios utilizan este software es un testimonio de su robustez y seguridad. Como el nombre implica, el vsftpd se diseñó desde cero para ser rápido, estable y muy seguro.

NOTA Como la mayoría de otros servicios, vsftpd es tan seguro como usted lo haga. Los autores del programa han proporcionado todas las herramientas necesarias para hacer el software tan seguro como sea posible al momento de la instalación, pero una mala configuración puede ocasionar que su sitio sea vulnerable. Recuerde comprobar dos veces su configuración y haga pruebas antes de ponerlo en marcha. También recuerde visitar el sitio Web de vsftpd con cierta regularidad para obtener actualizaciones del software.

En este capítulo trataremos la forma de obtener, instalar y configurar la última versión del vsftpd. Mostraremos cómo configurarlo para accesos privados así como para accesos anónimos. Y, por último, mostraremos cómo utilizar el cliente **ftp** a fin de probar su nuevo servidor FTP.

LA MECÁNICA DE FTP

El acto de transferir un archivo de una computadora a otra puede parecer trivial pero en realidad no lo es, cuando menos, no si lo está haciendo de manera correcta. En esta sección abordaremos los detalles de la interacción cliente/servidor FTP. Aunque esta información no es crucial para poner en marcha un servidor FTP, es importante saberlo para que pueda tomar decisiones sobre asuntos de seguridad y también para enfrentar problemas, sobre todo si necesita localizar fallas que no se manifiestan con claridad como fallas relacionadas con el FTP. (“¿Es un problema con la red, el servidor FTP, o el cliente FTP?”)

Interacciones cliente/servidor

El diseño original del FTP, concebido a inicios de la década de 1970, supuso algo que fue razonable por un largo periodo de tiempo acerca de Internet: los usuarios son una multitud amistosa.

Después de la comercialización de Internet alrededor de 1990-1991, ésta se hizo mucho más popular. Con la llegada de la World Wide Web (Telaraña Mundial), la población de usuarios de Internet y su popularidad creció aún más. Fue entonces cuando aparecieron problemas de seguridad que hasta ese momento eran relativamente desconocidos. Estos problemas de seguridad han hecho que el uso de firewalls sea un estándar en la mayoría de las redes.

El diseño original del FTP no se lleva muy bien que digamos con el ambiente hostil que hoy reina en Internet y que nos obliga a utilizar firewalls. En la misma medida que FTP facilita el intercambio de archivos entre un cliente FTP y un servidor FTP, su diseño presenta algunos pormenores integrados que merecen atención.

Uno de esos detalles deriva del hecho de que FTP utiliza dos puertos: uno de *control* (el puerto 21) y otro de *datos* (el puerto 20). El puerto de control sirve como un canal de comunicación entre el cliente y el servidor para el intercambio de comandos y respuestas, mientras que el puerto de datos se usa sólo para el intercambio de datos, los cuales pueden ser un archivo, parte de un archivo, o un listado de un directorio. FTP puede operar en dos modos: el modo *FTP activo* y el modo *FTP pasivo*.

FTP activo

El modo FTP activo se usó tradicionalmente en las especificaciones FTP originales. En este modo, el cliente se conecta desde un puerto efímero (número mayor que 1024) hasta el puerto de comandos del servidor FTP (el puerto 21). Cuando el cliente está listo para transferir datos, el servidor abre una conexión desde su puerto de datos (el puerto 20) hacia la combinación dirección IP y puerto efímero proporcionada por el cliente. La diferencia clave radica en que el cliente no hace la conexión de datos real hacia el servidor sino que le informa a éste sobre su propio puerto (emitiendo el comando PORT); el servidor entonces se conecta de regreso hacia el puerto especificado. El servidor puede ser considerado como la parte activa (o el investigador) en este modo FTP.

Desde la perspectiva de un cliente FTP que está detrás de un firewall, el modo FTP activo presenta un ligero problema: el firewall del lado del cliente quizás frunza la ceja (o niegue de plano) conexiones que se originan o inician desde Internet, desde un puerto de servicio privilegiado (esto es, el puerto 20) hacia un puerto de servicio no privilegiado en el cliente que se supone debe proteger.

FTP pasivo

El cliente FTP emite un comando PASV para indicar que quiere acceder datos en el modo pasivo, y el servidor entonces responde con una dirección IP y un número de puerto efímero propio hacia el cual el cliente puede conectarse con el fin de llevar a cabo la transferencia de datos. El comando PASV emitido por el cliente le dice al servidor que “escuche” en un puerto de datos que no es su puerto de datos normal (esto es, el puerto 20) y que espere a que se establezca una conexión en vez de que inicie una. La diferencia clave en este caso radica en que el cliente es el que inicia la conexión hacia el puerto y la dirección IP proporcionadas por el servidor. En este caso, el servidor puede ser considerado como la parte pasiva en la comunicación de datos.

Desde la perspectiva de un servidor FTP que está detrás de un firewall, el modo FTP pasivo es ligeramente problemático porque el instinto natural de un firewall sería negar conexiones que se originan desde Internet con destino hacia puertos efímeros de los sistemas a los que se supone debe proteger. Un síntoma típico de este tipo de comportamiento es cuando el cliente parece capaz de conectarse al servidor sin problema, pero la conexión parece colgarse cada vez que ocurre un intento de transferencia de datos.

Para lidiar con algunos de los asuntos pertinentes a FTP y firewalls, muchos de éstos implementan proxies para FTP a nivel aplicación, los cuales mantienen control de las solicitudes FTP y abren puertos altos cuando se necesitan para recibir datos desde un sitio remoto.

Obtención e instalación de vsftpd

El paquete vsftpd es el software para servidor FTP que se envía con varias de las distribuciones modernas de Linux. En particular, es el servidor FTP que viene con Fedora Core, RHEL, SuSE y demás. La última versión del software puede obtenerse desde el sitio Web oficial, <http://vsftpd.beasts.org>. El sitio Web también ofrece buena documentación y las últimas noticias acerca del software. Pero debido a que se trata de la solución de servidor FTP que se envía con Fedora Core, puede instalarlo con suma facilidad desde el medio de instalación o directamente desde cualquiera de los almacenes de software de Fedora Core.

En esta sección y la siguiente nos concentraremos en mostrar cómo se realiza la instalación y la configuración del software desde el binario preempacado, vía RPM. Además, abordaremos brevemente el proceso de compilación e instalación del software desde el código fuente.

Construcción de vsftpd desde el código fuente

La última versión que estaba disponible al momento de escribir esta obra era vsftpd-2.0.3.tar.gz. Este paquete puede obtenerse en forma directa desde <ftp://vsftpd.beasts.org/users/cevans/vsftpd-2.0.3.tar.gz>. Siga estos pasos para descargar y armar el programa:

1. Utilice el comando **wget** para descargar a toda velocidad el programa en el directorio **/usr/local/src**. Escriba

```
[root@serverA src]# wget ftp://vsftpd.beasts.org/users/cevans/vsftpd-2.0.3.tar.gz
```
2. Utilice el comando **tar** para decodificar y descomprimir el archivo **vsftpd-2.0.3.tar.gz** y luego entre al directorio **vsftpd** que se creó. Escriba

```
[root@serverA src]# tar xvzf vsftpd-2.0.3.tar.gz ; cd vsftpd-2.0.3
```
3. Destine unos minutos a estudiar los archivos **README** e **INSTALL** en el directorio.
4. Ejecute el comando **make** para compilar el software. Esto también dará por resultado el armado del ejecutable vsftpd en el raíz del directorio armado. Escriba

```
[root@serverA vsftpd-2.0.3]# make
```
5. Enseguida podrá ejecutar el comando **make install** para instalar en forma automática el vsftpd ejecutable y las páginas del manual (man) en los destinos predeterminados dentro de su sistema de archivos local (esto es, el vsftpd binario se instalará bajo **/usr/local/sbin/**). Escriba

```
[root@serverA vsftpd-2.0.3]# make install
```

Si quiere utilizar el archivo de configuración muestra **vsftpd.conf** tendrá que copiarlo a mano a un destino tal como el directorio **/etc**; de no hacerlo, puede generar uno desde cero.

Primero presentaremos el proceso de instalación del software desde un binario RPM:

1. Mientras mantiene abierta una sesión en el sistema como el superusuario (superuser), utilice el comando **up2date** para descargar e instalar en forma simultánea el vsftpd. Escriba (y luego presione y (yes) para responder afirmativamente cuando se le pregunte)

```
[root@serverA ~]# up2date -i vsftpd
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
Instalando...
2:vsftpd                                ##### [100%]
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
```

NOTA También puede descargar a mano el software desde alguno de los almacenes de Fedora Core en Internet, por ejemplo, desde <http://download.fedoraproject.org/pub/fedoraproject/4/i386/os/Fedora/RPMS/> vsftpd-2.0.3-1.i386.rpm. Y, como opción, puede instalarlo de manera directa desde el medio de instalación montado (CD o DVD). El software estará en el directorio /medio_de_instalación/Fedora/RPMS/.

2. Confirme que el software ha sido instalado. Escriba

```
[root@serverA ~]# rpm -q vsftpd
vsftpd-*
```

La configuración de vsftpd

Ahora que ya tiene instalado el software, el siguiente paso será configurarlo. El software vsftpd que se instaló en la sección anterior también instaló otros archivos y directorios en el sistema de archivo local. En la tabla 17-1 se presentan algunos de los archivos y directorios más importantes que aparecen instalados con el RPM vsftpd.

El archivo de configuración vsftpd.conf

Como se dijo antes, el archivo de configuración principal para el servidor FTP es **vsftpd.conf**. Cuando la instalación se realiza vía RPM, generalmente este archivo quedará en el directorio **/etc/vsftpd/**. Este archivo es muy fácil de administrar y entender, contiene pares de opciones (directrices) y valores que están en el siguiente formato simple

```
option=value
```

SUGERENCIA Es un error poner espacios entre la opción, el signo igual (=) y el valor.

Como sucede con otros archivos de configuración Linux/UNIX, los comentarios dentro del archivo se denotan por líneas que empiezan con el signo de número (#). Para ver el significado de cada una de las directrices, debería consultar la página **vsftpd.conf** en el manual utilizando el comando **man** como sigue:

```
[root@serverA ~]# man vsftpd.conf
```

Archivo	Descripción
/usr/sbin/vsftpd	Éste es el vsftpd ejecutable principal. Es el mismísimo demonio.
/etc/vsftpd/vsftpd.conf	Éste es el archivo de configuración principal para el demonio vsftpd . Contiene muchas directrices que controlan el comportamiento del servidor FTP.
/etc/vsftpd.ftpusers	Archivo de texto que almacena la lista de usuarios a los que <i>no</i> se les permite acceso al servidor FTP. Este archivo es referenciado por el sistema Pluggable Authentication Module (PAM) (Módulo de Autenticación Acoplable).
/etc/vsftpd.user_list	Archivo de texto utilizado ya sea para permitir o denegar acceso a los usuarios listados. Ello se logra según el valor de la directriz userlist_deny en el archivo vsftpd.conf .
/var/ftp	Éste es el directorio de trabajo del servidor FTP.
/var/ftp/pub	Este directorio sirve como el directorio que contiene archivos destinados al acceso anónimo al servidor FTP.

Tabla 17-1. Archivo de configuración y directorios de vsftpd

Las opciones (o directrices) en el archivo **/etc/vsftpd/vsftpd.conf** pueden organizarse en categorías según el rol que juegan. En la tabla 17-2 se tratan algunas de estas categorías.

NOTA Los valores posibles de las opciones en el archivo de configuración también se pueden dividir en tres categorías: las opciones **Booleanas** (por ejemplo, YES, NO), las opciones **Numéricas** (por ejemplo, 007, 770), y las opciones de **Cadena** (por ejemplo, raíz, /etc/vsftpd.chroot_list).

Inicio y prueba del servidor FTP

El demonio **vsftpd** está listo para ponerse en marcha apenas lo sacamos de la caja. Viene con algunos parámetros de configuración predeterminados que le permiten levantar el polvo de inmediato.

Desde luego, necesitamos iniciar el servicio. Después de ello, el resto de esta sección lo llevará por el proceso de pruebas del servidor FTP conectándose a éste desde un cliente FTP.

Tipo de opción	Descripción	Ejemplos
Opciones del demonio	Estas opciones controlan el comportamiento general del demonio vsftpd .	listen Cuando se habilita, vsftpd ejecutará en modo independiente en vez de hacerlo bajo un superdemonio como xinetd o inetd . vsftpd por sí mismo entonces se encargará de escuchar y manejar conexiones entrantes. El valor predeterminado es NO.
Opciones de socket	Están relacionadas con opciones de red y puertos.	listen_address Especifica la dirección IP en la cual vsftpd escucha para detectar conexiones de red. Esta opción no tiene valor predeterminado. anon_max_rate La tasa máxima permitida para transferencia de datos, en bytes por segundo, para clientes anónimos. El valor predeterminado es 0 (ilimitado).
Opciones de seguridad	Estas opciones controlan en forma directa la autorización o la negación de acceso al servidor; esto es, las opciones ofrecen un mecanismo de acceso de control integrado al servidor FTP.	listen_port Éste es el puerto que vsftpd escuchará para detectar conexiones FTP entrantes. El valor predeterminado es 21. pasv_enable Habilita o inhabilita el método PASV para obtener una conexión de datos. El valor predeterminado es YES. port_enable Habilita o inhabilita el método PORT para obtener una conexión de datos. El valor predeterminado es YES. anonymous_enable Controla si los accesos anónimos serán permitidos o no. En caso de ser habilitados, ambos nombres de usuario y anónimos se reconocen como accesos anónimos. El valor predeterminado es YES.

Tabla 17-2. Opciones de configuración para vsftpd

Tipo de opción	Descripción	Ejemplos
		<p>tcp_wrappers Suponiendo que vsftpd se compiló con soporte para tcp_wrappers, las conexiones entrantes pasarán a través del control de acceso tcp_wrappers. El valor predeterminado es NO.</p>
		<p>local_enable Controla si los accesos locales se permiten o no. En caso de ser permitidos, las cuentas de usuarios normales en /etc/passwd se pueden usar para obtener acceso. Predeterminado: NO.</p>
		<p>userlist_enable vsftpd cargará una lista de usuarios desde el archivo especificado por la directriz userlist_file. Si un usuario intenta obtener acceso utilizando un nombre en esta lista, su acceso se rechazará antes de que se le pida una contraseña. El valor predeterminado es NO.</p>
		<p>userlist_deny Esta opción se examina si la opción userlist_enable está activa. Cuando el valor de esta opción se ajusta en NO, entonces el acceso de usuarios se rechazará a menos que estén explícitamente listados en el archivo especificado por userlist_file. Cuando se niega el acceso, el rechazo se produce antes de que se le pida la contraseña al usuario; esto previene que los usuarios envíen texto plano (sin encriptación) por la red. El valor predeterminado es YES.</p>
		<p>userlist_file Esta opción especifica el nombre del archivo que será cargado cuando la opción userlist_enable está activa. El valor predeterminado es vsftpd.user_list.</p>

Tabla 17-2. Opciones de configuración para **vsftpd** (*cont.*)

Tipo de opción	Descripción	Ejemplos
Opciones de transferencia de archivos	Estas opciones atan a las transferencias de archivos hacia y desde el servidor FTP.	cmds_allowed Especifica una lista de comandos FTP permitidos. Sin embargo, después del acceso, los comandos disponibles siempre se permiten, esto es, USER, PASS, QUIT; otros comandos se rechazan, por ejemplo, cmds_allowed=PASV, RETR, QUIT. Esta opción no tiene valor predeterminado.
Opciones de directorio	Estas opciones controlan el comportamiento de los directorios ofrecidos por el servidor FTP.	download_enable Si se ajusta en NO, se les negará el permiso a todas las solicitudes de descarga. Valor predeterminado: YES. write_enable Esta opción controla si se permite o rechaza cualesquiera de los comandos que cambian el sistema de archivos. Estos comandos son: STOR, DELE, RNFR, RNTO, MKD, RMD, APPE y SITE. Valor predeterminado: NO. chown_uploads Esta opción cambia la propiedad de todos los archivos subidos en forma anónima al usuario especificado en la opción chown_username. Valor predeterminado: NO. chown_username Especifica el nombre del usuario que se asignará como propietario de los archivos subidos en forma anónima. Valor predeterminado: root.
		use_localtime Cuando se activa, vsftpd mostrará los listados de directorio con el tiempo de la zona horaria del sistema local. El comportamiento predeterminado es mostrar el tiempo en GMT; esto es, el valor predeterminado es NO.

Tabla 17-2. Opciones de configuración para vsftpd (*cont.*)

Tipo de opción	Descripción	Ejemplos
Opciones de registro cronológico	Estas opciones controlan cómo y dónde vsftpd registra información.	<p>hide_ids Cuando esta opción se habilita todos los listados de directorios mostrarán ftp al igual que el usuario y el grupo para todos los archivos. Valor predeterminado: NO.</p> <p>dirlist_enable Habilita o inhabilita la facultad de realizar listados de directorio. Si se ajusta en NO, aparecerá un error de negación de servicio cuando se intente llevar a cabo el listado de un directorio. Valor predeterminado: YES.</p> <p>vsftpd_log_file Esta opción especifica el archivo principal de registro vsftpd. El valor predeterminado es /var/log/vsftpd.log.</p> <p>xferlog_enable Esta opción le dice al software que mantenga un registro de todas las transferencias de archivos conforme éstas ocurren.</p> <p>syslogd_enable Si se habilita, entonces toda salida de registro que hubiese ido a /var/log/vsftpd.log se envía en sustitución al registro del sistema. El registro cronológico se realiza bajo la instalación FTPD.</p>

Tabla 17-2. Opciones de configuración para vsftpd (*cont.*)

Así que empecemos por establecer una sesión muestra de FTP anónimo. Antes de ello, iniciaremos el servicio FTP.

1. Inicie el servicio FTP. Escriba

```
[root@serverA ~]# service vsftpd start
Starting vsftpd for vsftpd: [ OK ]
```

SUGERENCIA Si el comando **service** no está disponible en su distribución de Linux, tal vez pueda controlar el servicio ejecutando en forma directa el script de control de ejecución. Por ejemplo, quizás pueda iniciar **vsftpd** emitiendo el comando

```
[root@serverA ~]# /etc/init.d/vsftpd start
```

2. Inicie el programa FTP cliente para la línea de comandos y establezca una conexión con el servidor local del FTP como un usuario anónimo. Escriba

```
[root@serverA ~]# ftp localhost  
Connected to localhost.localdomain.  
220 (vsFTPd 2.0.1)  
530 Please login with USER and PASS.  
530 Please login with USER and PASS.  
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
```

3. Proporcione el nombre del usuario FTP anónimo cuando se le pregunte; esto es, escriba **ftp**.

```
Name (localhost:root) : ftp  
331 Please specify the password.
```

4. Proporcione cualquier cosa cuando se le pregunte la contraseña.

```
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.
```

5. Utilice el comando FTP **ls** (o **dir**) para solicitar un listado de archivos en el directorio actual en el servidor FTP.

```
ftp> ls  
227 Entering Passive Mode (127,0,0,1,41,244)  
150 Here comes the directory listing.  
drwxr-xr-x    2 0          0          4096 Oct  4  2004 pub  
226 Directory send OK.
```

6. Utilice el comando **pwd** para mostrar su directorio de trabajo actual en el servidor FTP.

```
ftp> pwd  
257 "/"
```

7. Con el comando **cd**, intente cambiar a un directorio fuera del directorio de trabajo permitido para el FTP anónimo; por ejemplo, intente cambiarse al directorio **/boot** del sistema local. Recibirá un mensaje que le avisa que dicha operación no es posible.

```
ftp> cd /boot  
550 Failed to change directory.
```

8. Salga del servidor FTP utilizando el comando FTP **bye**.

```
ftp> bye  
221 Goodbye.
```

Enseguida intentaremos conectarnos al servidor FTP utilizando una cuenta del sistema local. Para esta sesión utilizaremos el nombre de usuario “yyang” que se creó en un capítulo anterior. Así que empecemos una sesión FTP muestra con la debida autenticación:

1. Inicie de nuevo el programa **ftp** cliente para la línea de comandos. Escriba

```
[root@serverA ~]# ftp localhost
Connected to localhost.localdomain.
220 (vsFTPd 2.0.1)
530 Please login with USER and PASS.
530 Please login with USER and PASS.
KERBEROS_V4 rejected as an authentication type
```

2. Introduzca **yyang** como el nombre del usuario cuando se le pregunte.

```
Name (localhost:root): yyang
```

3. Debe proporcionar la contraseña para el usuario yyang cuando se le pregunte.

```
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

4. Utilice el comando **pwd** para mostrar su directorio de trabajo actual en el servidor FTP. Notará que el directorio mostrado es el directorio de inicio para el usuario yyang.

```
ftp> pwd
257 "/home/yyang"
```

5. Con el comando **cd** intente cambiarse a un directorio fuera del directorio de inicio del usuario yyang; por ejemplo, intente cambiarse al directorio **/boot** del sistema local. Recibirá un mensaje que le avisa que dicha operación sí fue posible.

```
ftp> cd /boot
250 Directory successfully changed.
```

6. Salga del servidor FTP utilizando el comando **bye**.

```
ftp> bye
221 Goodbye.
```

Tal como se demostró en estos ejemplos de sesiones FTP, la configuración predeterminada del **vsftpd** en nuestro sistema Fedora Core de muestra permite lo siguiente:

- ▼ **Acceso FTP anónimo** Significa que cualquier usuario desde cualquier lugar puede registrarse en el servidor utilizando el usuario **ftp** (o anónimo) con cualquier cosa como contraseña.
- ▲ **Acceso de usuarios locales** Significa que a todos los usuarios válidos en el sistema local, con entradas en la base de datos de usuarios (el archivo **/etc/passwd**), se les permite registrarse en el servidor FTP utilizando su nombre de usuario y contraseña normales.

Personalización del servidor FTP

El comportamiento predeterminado de **vsftpd** quizá no sea el que usted quiera para su servidor FTP final. En esta sección haremos un recorrido por el proceso de personalización de algunas de las opciones del servidor FTP que sirven para ciertos escenarios.

Configuración de un servidor FTP sólo anónimo

Primero haremos las configuraciones necesarias en nuestro servidor FTP de manera que *no* permita acceso a usuarios que tengan cuentas regulares en el sistema. Este tipo de servidores es útil para sitios grandes que tienen archivos que deben ofrecer al público en general vía FTP. En tal escenario es impráctico crear una cuenta para cada usuario cuando éstos pueden ser miles.

Por fortuna, **vsftpd** está listo para funcionar como un servidor FTP anónimo cuando apenas termina de instalarse. No obstante, examinaremos las opciones de configuración en el archivo **vsftpd.conf** que aseguran esta función y también inhabilitan las opciones que se requieren.

Con el editor de textos de su preferencia, abra el archivo **/etc/vsftpd/vsftpd.conf** para su edición. Escudriñe el archivo y asegúrese de que cuando menos las directrices listadas enseguida estén presentes (si lo están pero están marcadas como comentarios, habrá que eliminar el símbolo de número [#] o cambiar el valor de la opción).

```
listen=YES
xferlog_enable=YES
anonymous_enable=YES
local_enable=NO
write_enable=NO
```

Encontrará que las opciones de la lista anterior son suficientes para poner en marcha su servidor FTP sólo anónimo, lo cual quizá le lleve a sobreescribir el archivo **/etc/vsftpd/vsftpd.conf** existente con las opciones mostradas. Esto ayudará a mantener el archivo mencionado más ligero y limpio.

SUGERENCIA Casi todos los sistemas Linux vienen configurados en forma predeterminada con un usuario “ftp”. Se supone que esta cuenta debe ser una cuenta de sistema sin privilegios y especialmente para ser utilizada para accesos tipo FTP anónimo. Necesitará que esta cuenta exista en su sistema a fin de que funcione el FTP anónimo. Para corroborarlo, utilice la herramienta **getent**. Escriba

```
[root@serverA ~]# getent passwd ftp
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
```

Si no obtiene una salida similar a la mostrada, puede crear con rapidez la cuenta para el sistema FTP con el comando **useradd**. Para crear un usuario “ftp” a modo, escriba

```
[root@serverA ~]# useradd -c "FTP User" -d /var/ftp -r -s /sbin/nologin ftp
```

Si tuvo que hacer modificaciones al archivo **/etc/vsftpd/vsftpd.conf**, necesitará reiniciar el servicio **vsftpd**. Escriba

```
[root@serverA ~]# service vsftpd restart
```

Si el comando **service** no está disponible en su distribución de Linux, tal vez pueda controlar el servicio ejecutando en forma directa la secuencia de control de arranque. Por ejemplo, quizá pueda reiniciar **vsftpd** emitiendo el comando

```
[root@serverA ~]# /etc/init.d/vsftpd restart
```

Configuración de un servidor FTP con usuarios virtuales

Los usuarios virtuales son usuarios que en realidad no existen; esto es, son usuarios que no tienen ningún privilegio ni funciones en el sistema además de aquellas para las que fueron creados. Esta particular disposición de un servidor FTP es un punto medio entre permitir el acceso a usuarios con cuentas en el sistema local y permitir sólo el acceso a usuarios anónimos. Si no hay manera de garantizar la seguridad de la conexión de la red desde el lado del usuario (cliente FTP) hasta el lado del servidor (servidor FTP), será temerario permitir a los usuarios con cuentas en el sistema local que se registraran en el servidor FTP. Esto se debe a que la transacción FTP entre ambas partes por lo general ocurre en texto plano. Esto, desde luego, es relevante ¡sólo si el servidor contiene cualquier dato de valor para los propietarios!

El empleo de usuarios virtuales permitirá a un sitio ofrecer contenido que debería estar disponible para usuarios no confiables pero aun hacer que el servicio también esté disponible para el público en general. En caso de que las credenciales de un usuario virtual (o de varios) fueran vulneradas, cuando menos podemos estar seguros de que los daños se mantendrán al mínimo.

SUGERENCIA También es posible configurar **vsftpd** para que realice la encriptación de comunicaciones entre sí y cualquiera de los clientes FTP mediante el uso de SSL. Éste es sumamente fácil de configurar pero es preciso considerar que los clientes FTP también deben trabajar con este tipo de comunicación, y, por desgracia, no hay muchos programas cliente FTP que incorporen esta funcionalidad. Si la seguridad es una preocupación seria, quizás quiera considerar el uso del programa **sftp** de OpenSSH para transferencias de archivos sencillas.

En esta sección vamos a crear dos usuarios virtuales llamados “ftp-usuario1” y “ftp-usuario2”. Estos usuarios no existirán en forma alguna en la base de datos de usuarios del sistema (el archivo **/etc/passwd**). Los siguientes pasos detallan el proceso:

1. Primero crearemos un archivo de texto plano que contendrá las combinaciones de nombre de usuario y contraseña para los usuarios virtuales. Cada nombre de usuario con su respectiva contraseña estará en líneas alternadas dentro del archivo. Por ejemplo, para el usuario ftp-usuario1, la contraseña será “usuario1”; y para el usuario ftp-usuario2, la contraseña será “usuario2”. Nombraremos al archivo como **plain_vsftpd.txt**. Utilice el editor de textos de su preferencia para crear el archivo. Aquí utilizaremos **vi**:

```
[root@serverA ~]# vi plain_vsftpd.txt
```

2. Introduzca este texto en el archivo:

```
ftp-usuario1  
usuario1  
ftp-usuario2  
usuario2
```

3. Guarde los cambios en el archivo y salga del editor de textos.

4. Convierta el archivo de texto plano que se creó en el paso 2 en un archivo con formato Berkeley DB (db) que pueda ser utilizado con la biblioteca pam_userdb.so. Guarde la salida en un archivo llamado **hash_vsftpd.db** almacenado bajo el directorio /etc. Escriba

```
[root@serverA ~]# db_load -T -t hash -f plain_vsftpd.txt  
/etc/hash_vsftpd.db
```

NOTA Necesita tener instalado el paquete **db4-utils** a fin de utilizar el programa **db_load**. Puede instalarlo rápidamente usando Yum con el comando **yum install db4-utils**. O búsqelo en el medio de instalación.

5. Restrinja el acceso al archivo de la base de datos de usuarios virtuales configurándole permisos más restrictivos. Esto asegurará que no pueda ser leído por cualquier usuario ajeno que está de paso en el sistema. Escriba

```
[root@serverA ~]# chmod 600 /etc/hash_vsftpd.db
```

6. Enseguida crearemos un archivo PAM que el servicio FTP pueda utilizar como el nuevo archivo de la base de datos de usuarios virtuales. Nombraremos el archivo como **virtual-ftp** y lo guardaremos debajo del directorio **/etc/pam.d/**. Utilice cualquier editor de textos para crear el archivo:

```
[root@serverA ~]# vi /etc/pam.d/virtual-ftp
```

7. Introduzca el siguiente texto en este archivo:

```
auth required /lib/security/pam_userdb.so db=/etc/hash_vsftpd  
account required /lib/security/pam_userdb.so db=/etc/hash_vsftpd
```

Estas entradas le dicen al sistema PAM que realice la autenticación de usuarios utilizando la nueva base de datos en el archivo **hash_vsftpd.db**.

8. Guarde los cambios en un archivo llamado **virtual-ftp** en el directorio **/etc/pam.d/**.
9. Lo que sigue es la creación de un ambiente de inicio para nuestros usuarios FTP virtuales. Haremos trampa y utilizaremos la estructura de directorios existente en el servidor FTP para crear una subcarpeta que almacenará los archivos que queremos que estén disponibles para los usuarios virtuales. Escriba

```
[root@serverA ~]# mkdir -p /var/ftp/private
```

SUGERENCIA Hicimos trampa en el paso 9 para evitarnos la tarea de realizar el proceso de creación de un usuario FTP visitante al que finalmente los usuarios virtuales se ligarán, y también para evitar distraernos con asuntos sobre permisos, ya que el sistema ya tiene un sistema de cuentas FTP a partir del cual podemos apalancarnos. Busque la directriz **guest_username** dentro de la página man de **vsftpd.conf** para obtener más información (**man vsftpd.conf**).

10. Ahora crearemos nuestro archivo **vsftpd.conf** personalizado que permitirá la existencia del esquema planeado en su totalidad. Con el editor de textos de su preferencia, abra el archivo **/etc/vsftpd/vsftpd.conf** para editarlo. Escudriñe el archivo y asegúrese de que cuando menos las directrices listadas enseguida estén presentes (si están presentes pero están marcadas como comentarios, habrá que eliminar el símbolo de número [#] o tam-

bien cambiar el valor de la opción). Añadimos comentarios para explicar las directrices menos obvias.

```
listen=YES
#NO queremos permitir que los usuarios accedan como anónimos
anonymous_enable=NO
xferlog_enable=YES
#Esto es para el servicio PAM que creamos y que fue nombrado virtual-ftp
pam_service_name=virtual-ftp
#Habilita el uso del archivo /etc/vsftpd.user_list
userlist_enable=YES
#NO debemos negar el acceso a usuarios especificados en el archivo /etc/vsftpd.user_list
userlist_deny=NO
userlist_file=/etc/vsftpd.user_list
tcp_wrappers=YES
local_enable=YES
#Esto activa los usuarios virtuales.
guest_enable=YES
#Mapea todos los usuarios virtuales al usuario real llamado "ftp"
guest_username=ftp
#El directorio ftp raíz en el servidor para todos los usuarios virtuales será /var/ftp/private/
local_root=/var/ftp/private/
```

SUGERENCIA Si decide no editar el archivo de configuración existente y crear uno desde cero, encontrará que las opciones sirven a nuestro propósito sin que nada adicional le haga falta. ¡El software vsftpd asumirá valores predeterminados interconstruidos para cualquier opción que no especifiquemos explícitamente en el archivo de configuración! Desde luego, puede omitir las líneas de comentarios para ahorrar algo de escritura.

11. Necesitaremos crear (o editar) el archivo `/etc/vsftpd.user_list` al que se hizo referencia en la configuración del paso 10. Para crear la entrada del primer usuario virtual, escriba

```
[root@serverA ~]# echo ftp-usuario1 > /etc/vsftpd.user_list
```

12. Para crear la entrada del segundo usuario virtual, escriba

```
[root@serverA ~]# echo ftp-usuario2 >> /etc/vsftpd.user_list
```

13. Ahora estamos listos para iniciar o reiniciar el servidor FTP. Escriba

```
[root@serverA ~]# service vsftpd restart
```

14. Ahora verificaremos que el servidor FTP esté comportándose de la manera esperada. Para ello, es necesario establecer una conexión como un usuario FTP virtual. Conéctese al servidor como `ftp-usuario1` (recuerde que la contraseña FTP para ese usuario es “`usuario1`”).

```
[root@serverA ~]# ftp localhost
Connected to localhost.localdomain.
220 (vsFTPd 2.0.1)
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
Name (localhost:root): ftp-usuario1
331 Please specify the password.
```

```
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> ls -l  
227 Entering Passive Mode (127,0,0,1,51,81)  
150 Here comes the directory listing.  
226 Directory send OK.  
ftp> pwd  
257 "/"  
ftp> cd /boot  
550 Failed to change directory.  
ftp> bye  
221 Goodbye.
```

15. También probaremos si los usuarios anónimos son, en efecto, rechazados por el servidor.

```
[root@serverA ~]# ftp localhost  
Connected to localhost.localdomain.  
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...  
Name (localhost:root): ftp  
530 Permission denied.  
Login failed.  
ftp> bye  
221 Goodbye.
```

16. Finalmente verificaremos que los usuarios locales (por ejemplo, el usuario Ying Yang) no pueden conectarse al servidor.

```
[root@serverA ~]# ftp localhost  
Connected to localhost.localdomain.  
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...  
Name (localhost:root): yyang  
530 Permission denied.  
Login failed.  
ftp> bye  
221 Goodbye.
```

RESUMEN

El demonio Very Secure FTP es un poderoso programa que ofrece todas las características que podría necesitar para mantener en funcionamiento y de manera segura un servidor FTP de uso comercial. En este capítulo abordamos el proceso de compilación, instalación y configuración del servidor vsftpd desde el código fuente y desde el binario RPM. En específico, cubrimos los siguientes temas:

- ▼ Algunas importantes opciones de configuración de uso común para vsftpd.
- Detalles sobre el protocolo FTP y sus efectos en firewalls.

- Puesta en marcha de servidores FTP anónimos.
- ▲ Puesta en marcha de un servidor FTP que sólo permite el acceso de usuarios virtuales.

Esta información es suficiente para mantener su servidor FTP ronroneando por un buen rato. Desde luego, como cualquier medio impreso acerca de software, este texto envejecerá y la información aquí contenida paulatinamente será obsoleta. Asegúrese de visitar de vez en cuando el sitio Web del vsftpd para aprender no sólo sobre los últimos desarrollos sino también sobre la documentación más reciente.

CAPÍTULO 18



Puesta en marcha
de un servidor Web
utilizando Apache

En este capítulo abordaremos el proceso de instalación y configuración del *servidor HTTP Apache* (<http://www.apache.org>) sobre un servidor Linux. Apache es software libre difundido bajo la licencia Apache. Según una respetada firma de análisis en la Red (Netcraft Ltd., <http://www.netcraft.co.uk>), Apache mantiene una participación de mercado superior a 50%. Este nivel de aceptación en la comunidad de Internet viene de las siguientes ventajas y beneficios proporcionados por el software de servidor Apache:

- ▼ Es estable.
- Varios sitios importantes como Amazon.com e IBM lo están utilizando.
- El programa y los componentes relacionados son de fuente abierta en su totalidad.
- Trabaja sobre una amplia gama de plataformas (todas las variantes populares de UNIX, algunas de las variantes no tan populares de UNIX y, por si esto fuera poco, hasta en Windows 2000, NT y 2003).
- Es extremadamente flexible.
- ▲ Ha probado ser seguro.

Antes de que nos adentremos en los pasos necesarios para configurar Apache, haremos un recorrido por los fundamentos del protocolo HTTP y también por algunas partes internas de Apache, como lo es su modelo de titularidad de procesos. Esta información le ayudará a comprender por qué Apache está configurado para trabajar de la forma como lo hace.

COMPRENSIÓN DEL PROTOCOLO HTTP

El Hypertext Transfer Protocol (HTTP) (Protocolo para Transferencia de Hipertexto) es, claro está, uno de los pilares de la World Wide Web, y Apache es la implementación servidor de dicho protocolo. Navegadores tales como Mozilla, Firefox y Microsoft Internet Explorer son implementaciones de clientes de HTTP.

Al momento de escribir esta obra, el protocolo HTTP iba en la versión 1.1 y estaba documentado en RFC 2616 (para obtener más detalles, consulte <http://www.ietf.org/rfc/rfc2616.txt>).

Encabezados

Cuando un cliente Web se conecta a un servidor Web, el método predeterminado del cliente para hacer esta conexión es contactar al puerto TCP 80 del servidor. Una vez conectado, el servidor Web se queda quieto; depende del cliente enviar comandos HTTP aprobados para hacer solicitudes al servidor. Con cada comando llega un *encabezado de solicitud* que incluye información acerca del cliente. Por ejemplo, cuando se utiliza el cliente de Mozilla sobre Linux, el servidor Web recibirá la siguiente información desde el cliente:

```
GET / HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/60.06 [en] (X11; U; Linux 2.6.12 i686)
Host: localhost:80
```

```
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,  
image/png, /  
Accept-Encoding: gzip  
Accept-Language: en  
Accept-Charset: iso-8859-1,* ,utf-8
```

La primera línea contiene el comando HTTP **GET**, el cual pide al servidor que entregue un archivo. El resto de la información conforma el encabezado, el cual informa al servidor acerca del cliente, el tipo de archivos que el cliente aceptará, y demás. Muchos servidores utilizan esta información para determinar lo que se puede y lo que no se puede enviar al cliente y para propósitos de registro cronológico.

Con el encabezado de solicitud se pueden enviar encabezados adicionales. Por ejemplo, cuando un cliente utiliza un hipervínculo para acceder al sitio del servidor, en el encabezado también aparece una entrada que muestra el sitio desde el cual se originó la solicitud del cliente.

Cuando recibe una línea vacía, el servidor sabe que el encabezado está completo. Una vez que se recibe el encabezado de solicitud, responde con el contenido solicitado, precedido por un encabezado del servidor. El encabezado del servidor proporciona al cliente información acerca del servidor, la cantidad de información que el cliente está a punto de recibir, y el tipo de datos que serán enviados. Por ejemplo, cuando se envía a un servidor HTTP el encabezado de solicitud antes mostrado, da por resultado el siguiente encabezado de respuesta del servidor:

```
HTTP/1.1 200 OK  
Date: Thu, 02 Jun 2005 14:03:31 GMT  
Server: Apache/2.0.52 (Fedora)  
Last-Modified: Thu, 02 Jun 2007 11:41:32 GMT  
ETag: "3f04-1f-b80bf300"  
Accept-Ranges: bytes  
Content-Length: 31  
Connection: close  
Content-Type: text/html; charset=UTF-8
```

Al encabezado de respuesta le siguen una línea vacía y luego el contenido real de la transmisión.

Puertos

El puerto predeterminado para solicitudes HTTP es el puerto 80, aunque también puede configurar un servidor Web para que utilice un puerto diferente (arbitrariamente seleccionado) que no use ningún otro servicio. Esto permite a un sitio tener en funcionamiento varios servidores Web en el mismo anfitrión, cada uno en diferente puerto. Algunos sitios utilizan este arreglo para configuraciones múltiples en sus servidores Web, brindando atención a diversos tipos de solicitudes de clientes.

Cuando un sitio ejecuta un servidor Web en un puerto no estándar, puede ver ese número de puerto en el URL del sitio. Por ejemplo, la dirección <http://www.redhat.com> con un número de puerto especificado se vería como <http://www.redhat.com:80>.

SUGERENCIA No cometa el error de procurar “seguridad con oscuridad”. Si su servidor está en un puerto TCP no estándar, eso no garantiza que los vándalos en Internet no lo encontrarán. Debido a la naturaleza automatizada de las herramientas existentes para ataque de sitios, se necesitan menos de 100 líneas de código en C para peinar un servidor y encontrar los puertos que están ejecutando servidores Web. El uso de puertos TCP no estandarizados no mantiene más seguro a su sitio.

Titularidad de procesos y seguridad

Como se discutió en capítulos anteriores, poner en funcionamiento un servidor Web bajo UNIX le obliga a lidiar con el modelo Linux (y UNIX en general). Tratándose de permisos, ello significa que cada proceso tiene un dueño y ese dueño tiene derechos limitados en el sistema.

Cuando se inicia un programa (proceso), hereda los permisos del proceso que lo creó. Por ejemplo, si usted se registró como el usuario raíz (root), el shell en el cual usted realiza su trabajo tiene todos aquellos derechos que tiene el raíz. Además, cualquier proceso que inicie desde dicho shell heredará todos los derechos del usuario raíz. Es posible restar derechos a los procesos pero no es posible otorgarles más.

NOTA Hay una excepción al principio de herencia de Linux. Los programas configurados con el bit SetUID no heredan los derechos de los procesos que los crean sino que empiezan con los derechos especificados por el propietario del archivo en sí. Por ejemplo, el archivo que contiene el programa **su** (**/bin/su**) pertenece al raíz y tiene el bit SetUID activado. Si el usuario *yyang* ejecuta el programa **su**, ese programa no heredará los derechos de *yyang* y en vez de ello iniciará con los derechos del superusuario (raíz).

Procesamiento de la titularidad en Apache

Para efectuar configuraciones de redes, el servidor HTTP Apache debe iniciar con permisos de usuario raíz. En específico, necesita ligarse a sí mismo con el puerto 80 de manera que pueda escuchar solicitudes y aceptar conexiones. Una vez que lo hace, Apache puede disminuir sus derechos y funcionar como un usuario no raíz, como se especifica en sus archivos de configuración. De manera predeterminada, éste es el usuario *nobody* (nadie).

Como usuario *nobody*, Apache puede leer solamente los archivos que el usuario *nobody* tiene permiso de leer. Así, si los permisos de un archivo se activan de manera que sólo los pueda leer su propietario, éste debe ser *nobody*. Para cualquier archivo que usted quiera poner a disposición del usuario *nobody* establezca los permisos de ese archivo para que sea legible por todo el mundo.

```
chmod a+r nombrearchivo
```

donde **nombrearchivo** es el nombre del archivo.

SUGERENCIA Las distribuciones más populares de Linux crean un usuario específico para Apache, y este usuario, en forma automática, se añadirá o creará en el sistema de la base de datos de usuarios si es que no existe con anterioridad. En SuSE Linux, el usuario predeterminado es “www”, y en Fedora Core y RHEL este usuario se llama “apache”.

La seguridad es especialmente importante para sitios que utilizan scripts CGI. Limitar los permisos del servidor Web reduce las posibilidades de que alguien pueda enviar solicitudes ma-

liciosas al servidor. Los procesos del servidor y los correspondientes scripts CGI pueden romper sólo aquello a lo que pueden acceder. Como usuario nobody, los scripts y procesos no tienen acceso a los mismos archivos clave a los que el usuario raíz tiene acceso (recuerde que el raíz puede acceder a todo, sin importar los permisos).

NOTA En caso de que decida permitir scripts CGI en su servidor, ponga especial atención en la manera en que están escritos. Asegúrese de que no sea posible que alguna entrada que viene sobre la red haga que el script CGI ejecute algo que no debe. Aunque no hay estadísticas al respecto, la mayoría de los ataques a sitios Web se deben a servidores Web mal configurados y/o scripts CGI escritos deficientemente.

INSTALACIÓN DEL SERVIDOR HTTP APACHE

Las distribuciones modernas de Linux vienen acompañadas de un paquete binario del software para servidor HTTP Apache en formato RPM, así que instalar el software es por lo general tan sencillo como utilizar el comando **rpm**. Esta sección lo lleva por el proceso de obtención e instalación del programa vía RPM. También se menciona la instalación del software a partir del código fuente si es que elige esa ruta. La configuración real del servidor que se trata en secciones posteriores se aplica a ambas clases de la instalación (desde la fuente y desde el RPM).

Hay varias maneras de obtener el RPM de Apache. He aquí algunas de ellas:

- ▼ Descargue el RPM de Apache para su sistema operativo desde el almacén de distribución del software. Para Fedora Core, puede obtener una copia del programa en http://download.fedoraproject.org/pub/fedoraproject.org/linux/core/4/i386/os/Fedora/RPMS/httpd-2*.rpm.
- Puede instalarlo desde el medio de instalación, desde el directorio **/Fedora/RPMS**.
- ▲ Puede descargar e instalar el programa en forma directa desde un almacén utilizando el programa Yum. Ésta es quizás la manera más rápida si tiene una conexión activa a Internet. Y ésta es la manera que utilizaremos aquí.

Para utilizar el programa Yum, escriba

```
[root@serverA ~]# yum install httpd
```

Escriba **y** cuando se le pregunte que confirme si quiere instalar el programa y cualquiera de sus dependencias.

Para confirmar que el software en verdad quedó instalado, escriba

```
[root@serverA ~]# rpm -q httpd  
httpd-2.0.*
```

¡Y eso es todo! Ahora tiene instalado Apache.

Instalación de Apache desde el código fuente

En caso de que no le agraden los elementos predeterminados que están interconstruidos en el paquete RPM y quiera construir su software para servidor Web desde cero, puede obtener la última versión más estable del programa directamente del sitio Web apache.org. Enseguida presentamos un procedimiento para construirlo desde el código fuente.

Primero descargaremos el último código fuente del programa en el directorio **/usr/local/src/** desde el sitio Web apache.org. Puede utilizar el programa **wget** para hacerlo. Escriba

```
[root@serverA src]# wget http://www.apache.org/dist/httpd/httpd-2.0.54.tar.gz
```

Extraiga el archivo tar. Luego entre al directorio que se creó durante la extracción.

```
[root@serverA src]# tar -xvzf httpd-2.0.54.tar.gz  
[root@serverA src]# cd httpd-2.0.54/
```

Suponiendo que queremos que el programa del servidor Web sea instalado en el directorio **/usr/local/httpd/**, ejecutaremos el script **configure** con la opción **prefix** apropiada.

```
[root@serverA httpd-2.0.54]# ./configure --prefix=/usr/local/httpd
```

Ahora, ejecute **make**.

```
[root@serverA httpd-2.0.54]# make
```

Genere el directorio de trabajo del programa (esto es, **/usr/local/httpd/**) y luego ejecute **make install**.

```
[root@serverA httpd-2.0.54]# make install
```

Una vez que el comando **install** termina con éxito, se creará una estructura de directorio bajo **/usr/local/httpd/**, misma que contendrá los binarios, los archivos de configuración, los de registro cronológico, y demás, para el servidor Web.

Los módulos de Apache

Parte de lo que hace al servidor Apache tan poderoso y flexible es un diseño que permite extensiones mediante módulos. De fábrica, Apache viene con varios módulos que se instalan automáticamente durante la instalación predeterminada.

Si puede imaginar alguna función en especial, casi puede estar seguro de que alguien en algún lugar probablemente ya ha escrito un módulo que hace lo que usted imaginó para el servidor Web Apache. El módulo API de Apache está bien documentado, así que si tiene inclinaciones (y sabe cómo hacerlo), tal vez podría crear su propio módulo para Apache con el fin de proporcionarle la funcionalidad que usted requiere.

Para darle una idea del tipo de cosas que la gente está creando con módulos, visite <http://modules.apache.org>. Ahí podrá encontrar información sobre cómo extender las capacidades de Apache utilizando módulos. Algunos de los módulos Apache más comunes son

- ▼ **mod_cgi** Permite la ejecución de scripts CGI en el servidor Web.
- **mod_perl** Integra un intérprete Perl en el servidor Web Apache.
- **mod_aspdotnet** Proporciona una interfaz de anfitrión ASP.NET para la máquina ASP.NET de Microsoft.
- **mod_authz_ldap** Provee soporte para la autenticación de usuarios del servidor HTTP Apache contra una base de datos LDAP.
- **mod_ssl** Brinda criptografía pesada para el servidor Web Apache mediante los protocolos Secure Sockets Layer (SSL) y Transport Layer Security (TLS).
- **mod_ftp** Faculta al servidor Apache para que acepte conexiones FTP.
- ▲ **mod_userdir** Permite que se ofrezca contenido del usuario vía HTTP desde directorios específicos de usuarios en el servidor Web.

Si conoce el nombre de algún módulo en particular que usted quiera (y si el módulo es suficientemente popular), quizás encuentre que dicho módulo ya ha sido empacado en formato RPM de manera que podrá instalarlo utilizando los métodos RPM usuales. Por ejemplo, si quiere incluir el módulo SSL (**mod_ssl**) en la instalación de su servidor, en un sistema Fedora Core puede emitir el siguiente comando Yum para descargar e instalar en forma automática dicho módulo:

```
[root@serverA ~]# yum install mod_ssl
```

En vez de ello, puede visitar el sitio Web para proyectos de módulos Apache y buscar, descargar, compilar e instalar el módulo que requiera.

SUGERENCIA ¡Asegúrese de que el usuario *nobody* esté ahí! Si construye Apache desde el código fuente, el archivo de configuración muestra (**httpd.conf**) espera que el servidor Web se ejecute como el usuario *nobody* (nadie). Aunque dicho usuario existe en casi todas las distribuciones de Linux, si algo falla en el camino, quizás quiera revisar la base de datos de usuarios (**/etc/passwd**) para asegurarse de que el usuario *nobody* en verdad existe.

INICIO Y APAGADO DE APACHE

Una de las características más agradables de Linux es su habilidad para iniciar y detener servicios del sistema sin necesidad de reiniciar. Ello es fácil de realizar en el servidor Apache.

Para iniciar Apache en un sistema Fedora Core o cualquier otro sistema Red Hat, utilice este comando:

```
[root@serverA ~]# service httpd start
```

Para detener Apache, introduzca este comando:

```
[root@serverA ~]# service httpd stop
```

Después de hacer un cambio de configuración en el servidor Web que requiera que usted reinicie Apache, escriba

```
[root@serverA ~]# service httpd restart
```

SUGERENCIA En un sistema que ejecuta SuSE Linux, los comandos para iniciar y detener el servidor Web, respectivamente, son

```
[serverA ~]# rcapache2 start
```

y

```
[serverA ~]# rcapache2 stop
```

Inicio de Apache durante el arranque del sistema

Después de instalar el servidor Web, si decide que este servicio lo debe ofrecer su sistema todo el tiempo, necesitará realizar algunas configuraciones para que dicho servicio inicie en forma automática siempre que arranque el equipo. Es muy fácil olvidar esta configuración en un sistema que ha estado funcionando por mucho tiempo, sin necesidad de reiniciar. Pero si tiene que apagarlo debido a una falla que lo requiera, estará sorprendido y se preguntará por qué el servidor Web, que había funcionado a la perfección, sin incidentes, de pronto falla y no inicia al encender el sistema. A fin de evitar esta situación, es una buena idea tomarse unos minutos para llevar a cabo este proceso en las etapas tempranas de la configuración del servicio.

La mayoría de las versiones de Linux tienen disponible la utilidad **chkconfig**, la cual se puede usar para controlar cuáles servicios inician y en qué niveles de ejecución.

Para ver los niveles de ejecución con los que el servidor Web está configurado para iniciar, escriba

```
[root@serverA ~]# chkconfig --list httpd
httpd           0:off    1:off    2:off    3:off    4:off    5:off    6:off
```

Esta respuesta muestra que el servidor Web no está configurado para iniciar en ningún nivel de ejecución cuando la instalación ha sido recién terminada. Para cambiar esto y hacer que Apache inicie en forma automática en los niveles de ejecución 2, 3, 4 y 5, escriba

```
[root@serverA ~]# chkconfig httpd on
```

NOTA Sólo en caso de que usted esté trabajando en una versión de Apache que instaló a partir del código fuente, debe considerar que la utilidad **chkconfig** no sabrá acerca de los scripts de inicio y apagado para su servidor Web a menos que explícitamente le diga a la utilidad acerca de ello. Y, como tal, tendrá que recurrir a algunos otros trucos para configurar el sistema anfitrión de tal manera que traiga en forma automática a la vida el servidor Web cuando reinicie el equipo. Puede traer fácilmente otro script de arranque de otro sistema que ya esté trabajando (por lo general del directorio **/etc/init.d/**) y modificarlo para reflejar trayectorias correctas (por ejemplo, **/usr/local/httpd/** para personalizar su configuración. Los scripts existentes tentativamente deben llamarse **httpd** o **apache**.

PRUEBAS A SU INSTALACIÓN

Es posible realizar una prueba rápida de la instalación de su servidor Apache utilizando la página de inicio predeterminada. Para llevar a cabo esta prueba, primero utilice el siguiente comando para confirmar que el servidor Web está en operación:

```
[root@serverA ~]# service httpd status  
httpd (pid 26084 26080 26079 26078 26077 26074) is running....
```

En el sistema Fedora Core con el que hemos venido trabajando estos ejemplos, Apache tiene una página preestablecida que muestra a los visitantes en ausencia de una página de inicio predeterminada (por ejemplo, **index.html** o **index.htm**). El archivo que se muestra a los visitantes cuando no hay una página de inicio predeterminada es **/var/www/error/noindex.html**.

SUGERENCIA Si está trabajando con una versión de Apache que construyó a partir del código fuente, el directorio de trabajo desde el cual las páginas Web se ofrecen es <PREFIJO>/htdocs. Por ejemplo, si su prefijo de instalación es **/usr/local/httpd/**, entonces las páginas Web estarán dentro del directorio **/usr/local/httpd/htdocs/**.

A fin de saber si la instalación de su servidor Apache ocurrió sin problemas, inicie un navegador Web y visítelo en su máquina. Para hacer esto, tan sólo escriba **http://localhost** en la barra de direcciones de su navegador. Debería ver una página que le informa algo así como "Su servidor Apache HTTP está funcionado correctamente en este sitio". Si no puede verla, recorra los pasos de instalación de su servidor Apache y asegúrese de que no encontró errores en el proceso.

CONFIGURACIÓN DEL SERVIDOR APACHE

Apache soporta una amplia gama de opciones de configuración que son razonables y fáciles de seguir. Esto hace que poner en funcionamiento el servidor Web con varias configuraciones sea una tarea sencilla.

Esta sección lo lleva a través de una configuración básica. La configuración predeterminada, de hecho, es bastante buena y (lo crea o no) funciona apenas se instala, así que, si es aceptable para usted, no espere más y ¡pongase a crear sus documentos HTML! Apache permite varias personalizaciones comunes. Después de crear una página Web sencilla, mostraremos cómo puede hacer dichas personalizaciones en los archivos de configuración del servidor Apache.

Creación de una sencilla página de nivel raíz

Si lo desea, puede comenzar a añadir archivos a Apache desde ahora en el directorio **/var/www/html** que sirve para páginas de primer nivel (para una instalación a partir de código fuente, el directorio sería **/usr/local/httpd/htdocs**). Todos los archivos colocados en ese directorio deben ser legibles a escala mundial.

Como se mencionó antes, **index.html** es la página Web predeterminada de Apache. Veamos con mayor detenimiento la creación y el cambio de la página de inicio que viene de fábrica de manera que ahora lea “Bienvenido al servidorA.ejemplo.org”. He aquí los comandos:

```
[root@serverA ~]# cd /var/www/html/
[root@serverA html]# echo "Bienvenido al servidorA.ejemplo.org" >> index.html
[root@serverA html]# chmod 644 index.html
```

También podría utilizar un editor de texto como **vi** o **pico** para editar el archivo **index.html** y hacerlo más interesante.

Archivos de configuración de Apache

En sistemas Fedora Core o RHEL, estos archivos están ubicados en el directorio **/etc/httpd/conf/**, y para el ejemplo que derivó de la instalación desde código fuente, la ruta sería **/usr/local/httpd/conf/**. El archivo principal de configuración se llama **httpd.conf**.

La mejor manera de aprender más acerca de los archivos de configuración es leer el archivo **httpd.conf**. Este archivo tiene comentarios en abundancia que explican cada entrada, su papel, y los parámetros que puede utilizar.

Opciones de configuración comunes

Los ajustes predeterminados de configuración trabajan muy bien apenas culmina la instalación del software Apache, pues atienden necesidades básicas y es posible que no requieran modificaciones posteriores. Sin embargo, los administradores de sitio quizás necesiten personalizar su servidor o su sitio Web un poco más.

Esta sección aborda algunas directrices u opciones comunes utilizadas en el archivo de configuración de Apache.

ServerRoot

Utilizada para especificar el directorio base del servidor Web. En sistemas Linux Fedora y RHEL, el valor predeterminado es el directorio **/etc/httpd/**.

Sintaxis: **ServerRoot ruta-directorio**

Listen

Sirve para definir el puerto en el cual el servidor escucha solicitudes de conexión. También puede usarse para especificar direcciones IP particulares para las cuales el servidor Web acepta conexiones. El valor predeterminado para esta directriz es 80 para comunicaciones Web no seguras.

Sintaxis: **Listen [dirección-IP:] número-puerto**

ServerName

Esta directriz define el nombre del anfitrión y el puerto que el servidor utiliza para identificarse a sí mismo. En muchos sitios, los servidores cubren múltiples propósitos. Un servidor Web que no está bajo uso pesado, por ejemplo, probablemente debería compartir su asignación de

poder de cómputo con otro servicio. En tal caso, un nombre de computadora tal como “www” (FQDN=www.ejemplo.org) no sería una buena elección porque sugiere que la máquina tiene un solo propósito.

Es mejor dar un nombre neutral a un servidor para luego establecer entradas DNS CNAME o entradas múltiples para nombres de anfitrón en el archivo `/etc/hosts`. En otras palabras, puede darle al sistema varios nombres para acceder al servidor pero éste necesita saber sólo su verdadero nombre. Considere el caso de un servidor cuyo nombre de anfitrón es dioxina.eng.ejemplo.org que, además, fuese un servidor Web. Podría pensar en asignarle www.ventas.ejemplo.org como su alias de nombre de anfitrón. Sin embargo, como dioxina sabe que su nombre es dioxina, los usuarios que visiten el sitio Web www.ventas.ejemplo.org podrían confundirse al ver en sus navegadores que el verdadero nombre es dioxina.

Apache proporciona una manera de rodear este problema mediante la directriz **ServerName**. Esto le permite especificar lo que usted quiere que Apache presente como nombre de anfitrón para el servidor Web que verán los programas cliente de los visitantes.

Sintaxis: `ServerName nombre-dominio-completamente-calificado [:número-puerto]`

ServerAdmin

Ésta es la dirección de correo electrónico que el servidor incluye en los mensajes de error enviados al cliente.

Es una buena idea utilizar un alias de correo electrónico para el administrador del sitio Web debido a dos razones. Primero, puede haber más de un administrador. El uso de un alias permite expandirlo a una lista con varias direcciones de correo electrónico. Segundo, si el administrador actual se va, usted no querrá verse obligado a cambiar docenas de páginas Web y cambiar el nombre del administrador del sitio.

Sintaxis: `ServerAdmin dirección-correo-electrónico`

DocumentRoot

Define el directorio principal en el servidor Web desde el cual los archivos HTML se ofrecerán a solicitud de los programas cliente. En la mayoría de los sistemas Linux, `/var/www/html/` es el valor predeterminado para esta directriz.

SUGERENCIA En un servidor Web que se supone debe hospedar contenido Web en abundante cantidad, el sistema de archivos en el cual reside el directorio especificado debe contener a su vez suficiente espacio.

MaxClients

Establece un límite al número de solicitudes concurrentes que serán atendidas por el servidor Web.

LoadModule

Se utiliza para cargar o añadir módulos en la configuración actual de Apache. Añade el módulo especificado a la lista de módulos activos.

Sintaxis: `LoadModule nombre-módulo nombre-archivo`

User

Especifica la ID (identificación) del usuario que utilizará el servidor Web para responder solicitudes. El proceso del servidor inicia como el usuario raíz pero más tarde reduce sus privilegios a los del usuario especificado con esta opción. Este usuario debe tener sólo aquellos privilegios suficientes para acceder archivos y directorios que se pretende que sean vistos por el mundo exterior mediante el servidor Web. Asimismo, el usuario no debe contar con la capacidad de ejecutar código que no sea el relacionado con el HTTP o la Web.

En un sistema Fedora, el valor para esta directriz se instaura en forma automática como el usuario “apache”. En SuSE Linux, el valor se establece como el usuario “www”.

Sintaxis: `User idusuario-unix`

Group

Especifica un nombre para el grupo de procesos del servidor HTTP Apache. Es el grupo con el cual el servidor responderá a solicitudes. El valor predeterminado en las variantes Fedora y RHEL de Linux es “apache”. En SuSE Linux, el valor queda como el grupo “www”.

Sintaxis: `Group grupo-unix`

Include

Esta directriz permite que Apache especifique e incluya otros archivos de configuración en el momento de ejecución. Es útil sobre todo para propósitos de organización; por ejemplo, puede elegir que las directrices de configuración para diferentes dominios virtuales queden almacenadas en archivos bien nombrados de manera que Apache, en forma automática, sepa que tiene que incluirlos en el momento de ejecución.

Sintaxis: `Include nombre_archivo_to_include_o_ruta_directorio_to_include_`

UserDir

Esta directriz define el subdirectorio dentro del directorio de inicio de cada usuario donde los usuarios pueden colocar contenido personal que quieren compartir vía el servidor Web. Es usual que este directorio se llame **public_html** y también es usual que se almacene bajo el directorio de inicio de usuario. Esta opción depende, por supuesto, de la disponibilidad del módulo **mod_userdir** en la configuración del servidor Web.

Un ejemplo del uso de esta opción en el archivo **httpd.conf** es

```
UserDir publico_html
```

ErrorLog

Define la ubicación del archivo donde se registrarán los errores del servidor Web.

Sintaxis: `ErrorLog ruta_archivo|syslog[:facility]`

Ejemplo: `ErrorLog /var/log/httpd/error_log`

Guía rápida para publicar contenido HTTP desde los directorios de usuarios

Después de habilitar la opción **UserDir**, con los pasos siguientes el usuario yyang puede, si así lo quiere, publicar contenido Web desde su directorio de inicio mediante el servidor Web.

1. Mientras está registrado en el sistema como el usuario yyang, genere la carpeta **public_html**.

```
[yyang@serverA ~]# mkdir ~/public_html
```

2. Ajuste los permisos apropiados para la carpeta superior.

```
[yyang@serverA ~]# chmod a+x .
```

3. Ajuste los permisos pertinentes para la carpeta **public_html**.

```
[yyang@serverA ~]# chmod a+x public_html
```

4. Genere una página ejemplo llamada **index.html** dentro de la carpeta **public_html**.

```
[yyang@serverA ~]# echo "Página de inicio de Ying Yang" >> ~/public_html/index.html
```

Como resultado de estos comandos, los archivos colocados en el directorio **public_html** de un usuario en particular que sean ajustados para su lectura a escala mundial estarán en la Web mediante el servidor Web.

Para acceder al contenido de esa carpeta vía HTTP tendría que pedir a un navegador Web que se asome a un URL como el siguiente:

`http://<NOMBRE_DEL_ANFITRIÓN>/~<NOMBRE_DEL_USUARIO>`

donde **NOMBRE_DEL_ANFITRIÓN** es un nombre de dominio completamente calificado o la dirección IP del servidor Web. Y si está trabajando en el servidor Web, reemplace esta variable con **localhost**.

Para el ejemplo mostrado aquí con el usuario yyang, el URL exacto será `http://localhost/~yyang`.

SUGERENCIA En un sistema Fedora Core con el subsistema SELinux activado, quizá necesite hacer algo más para hacer que la directriz **UserDir** funcione. Ello se debe a los contextos de seguridad predeterminados para los archivos almacenados en el directorio de inicio de cada uno de los usuarios. El contexto predeterminado es **user_home_t**. A fin de que esta característica funcione correctamente, deberá cambiar el contexto de todos los archivos bajo `~/nombreusuario/public_html/` a **httpd_sys_content_t**. Esto permite que Apache lea los archivos en el directorio **public_html**. El comando es

```
[yyang@serverA ~]# chcon -Rt httpd_sys_content_t public_html/
```

LogLevel

Esta opción ajusta el nivel de verbosidad para los mensajes enviados a los registros cronológicos de error. Los niveles aceptados son emerg (emergencia), alert (alerta), crit (crítico), error, warn (advertencia), notice (aviso), info y debug (depuración). El nivel predeterminado es "warn".

Sintaxis: LogLevel level

Alias

Esta directriz permite que los documentos que constituyen el contenido Web se almacenen en cualquier otra ubicación del sistema de archivos diferente de la ubicación especificada por la directriz **DocumentRoot**. También permite definir abreviaturas (o alias) para rutas que de otra manera serían muy largas.

Sintaxis: Alias ruta_URL ruta_hacia_archivo_o_directorio

ScriptAlias

Esta opción especifica un archivo o directorio destino que contiene scripts CGI que se espera que sean procesados por el módulo CGI (**mod_cgi**).

Sintaxis: ScriptAlias ruta_URL ruta_hacia_archivo_o_directorio

Ejemplo: ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"

VirtualHost

Una de las características más utilizadas de Apache es su habilidad para mantener anfitriones virtuales. Esta opción hace posible que un solo servidor Web hospede varios sitios, como si cada uno tuviera su propio hardware dedicado, y permite que el servidor Web proporcione contenidos diferentes y autónomos, según el nombre del anfitrión, el número de puerto, o la dirección IP que solicita el cliente. Ello es posible gracias a que el protocolo HTTP 1.1 especifica el sitio deseado en el encabezado HTTP en vez de confiar en que el servidor aprenderá qué sitio entrega según esa dirección IP.

Esta directriz en realidad consta de dos etiquetas: una etiqueta de apertura `<VirtualHost>` y una de cierre `</VirtualHost>`. Se usa para especificar las opciones pertinentes a un anfitrión virtual en particular. La mayoría de las directrices tratadas anteriormente también son válidas aquí.

Sintaxis: `<VirtualHost dirección_IP_o_nombreanfitrión[:puerto] >`
`Opciones`
`</VirtualHost>`

Suponga, por ejemplo, que queremos poner en marcha la configuración para un anfitrión virtual llamado `www.otro-ejemplo.org`. Para hacer esto, podemos crear una entrada **VirtualHost** en el archivo **httpd.conf** (o utilice la directriz **Include** para especificar un archivo separado) como sigue:

```
<VirtualHost www.otro-ejemplo.org>
  ServerAdmin webmaster@otro-ejemplo.org
  DocumentRoot /www/docs/otro-ejemplo.org
```

```
ServerName www.otro-ejemplo.org
ErrorLog logs/otro-ejemplo.org-error_log
</VirtualHost>
```

Tome en cuenta que este pequeño ejemplo no muestra los comandos suficientes para configurar un anfitrión virtual utilizando la directriz **VirtualHost**: por resaltar un detalle, podemos mencionar que el valor de la opción **ServerName** arriba debe ser un nombre que pueda ser resuelto vía DNS (o cualquier otro método) hacia la máquina del servidor Web.

NOTA Las opciones y directrices de Apache son muy numerosas como para ser tratadas en esta sección. Sin embargo, el software viene con una amplio manual en línea, escrito en HTML, de tal manera que puede verlo en un navegador. Si instaló el software vía RPM, encontrará que dicha documentación fue empacada en un binario RPM aparte y, como resultado, tendrá que instalar el paquete apropiado (por ejemplo, `httpd-manual`) a fin de tener acceso a éste. Si descargó y construyó el software desde el código fuente, encontrará la documentación en el directorio **manual** dentro del prefijo de instalación (por ejemplo, `/usr/local/httpd/manual`). La documentación de Apache también está disponible en español, dentro del sitio Web del proyecto: <http://httpd.apache.org/docs/2.0/es>.

LOCALIZACIÓN DE PROBLEMAS EN APACHE

El proceso de cambiar configuraciones (o, inclusive, la instalación inicial) puede no ser tan sencilla como uno podría esperar. Afortunadamente, Apache hace un excelente trabajo al anotar en su registro cronológico de errores el motivo por el cual falló o la razón por la cual está fallando.

El archivo de registro cronológico de errores está ubicado en el directorio **logs**. Si está ejecutando una instalación estándar de Fedora Core o RHEL, este directorio es `/var/log/httpd/`. Si instaló Apache por su cuenta utilizando el método de instalación descrito antes en este capítulo, los registros cronológicos se ubican en el directorio `/usr/local/httpd/logs/`. En estos directorios encontrará dos archivos: **access_log** y **error_log**.

El archivo **access_log** no es otra cosa que un registro de aquellos archivos que han sido leídos por las personas que visitan su(s) sitio(s) Web. Contiene información acerca de si se completó con éxito la transferencia de archivos, desde dónde se originó (dirección IP) la solicitud, cuántos datos se transfirieron y a qué hora ocurrió la transferencia. Éste es un medio muy poderoso para determinar el uso de su sitio.

El archivo **error_log** contiene todos los errores que ocurren en Apache. Note que no todos los errores que ocurren son fatales, algunos son simples problemas con la conexión de un cliente, de lo cual Apache puede recuperarse para seguir en operación. Sin embargo, si ya inició Apache pero no puede visitar su propio sitio Web, entonces déle un vistazo a este registro cronológico para ver por qué Apache no está respondiendo. La forma más sencilla de ver los mensajes de error más recientes es mediante el uso del comando **tail**, como sigue:

```
[root@serverA html]# tail -n 10 /var/log/httpd/error_log
```

Si necesita ver más información del registro cronológico que eso, cambie el número 10 al número de líneas que necesite ver. Y si requiere ver los errores o registros conforme éstos ocurren en tiempo real, deberá utilizar la opción **-f** para el comando **tail**. Esto proporciona una herramienta

ta de depuración muy valiosa, pues es posible intentar nuevas cosas con el servidor (tal como solicitar páginas Web o reiniciar Apache) y ver los resultados de sus experimentos en una ventana de terminal virtual separada. En seguida se muestra el comando **tail** con el parámetro **-f**:

```
[root@serverA html]# tail -f /var/log/httpd/error_log
```

Este comando seguirá los registros cronológicos hasta que termine el programa (utilizando CTRL-C).

RESUMEN

En este capítulo cubrimos el proceso de poner en funcionamiento, desde cero, su propio servidor Web utilizando Apache. Este capítulo, por sí solo, es suficiente para permitirle publicar una página de un solo nivel y una configuración básica.

Es muy recomendable que se tome el tiempo necesario para ojear el manual de Apache. Está bien escrito, es conciso y flexible en grado suficiente como para que pueda configurar casi cualquier configuración que pueda imaginar.

Además de la documentación recomendada, se han escrito varios libros provechosos acerca de Apache. Por ejemplo, Apache: *The Definitive Guide, Third Edition* (O'Reilly, 2002) de Ben Laurie y Meter Laurie, cubre los detalles de Apache a profundidad. Dicha obra se enfoca en Apache y nada más que en Apache, así que no tendrá que pasar por cientos de páginas hasta encontrar lo que necesita.

CAPÍTULO 19



SMTP

El Simple Mail Transfer Protocol (SMTP) (Protocolo de transferencia de correo simple) es el estándar de mayor aceptación para el transporte de correo electrónico en Internet. Cualquiera que desee tener un servidor capaz de enviar y recibir correo a través de Internet debe estar preparado para manejarlo. Muchas redes internas también han adoptado el SMTP para sus servicios de correo privado gracias a su independencia de plataforma y su disponibilidad en todos los sistemas operativos populares. En este capítulo primero discutiremos la mecánica del SMTP como protocolo y su relación con otros protocolos relacionados tales como POP e IMAP. Luego abordaremos algunos detalles sobre el servidor Postfix SMTP, uno de los más sencillos y seguros.

COMPRENSIÓN DE SMTP

El protocolo SMTP define el método mediante el cual el correo electrónico se envía de un anfitrión a otro. Nada más. No define cómo debe almacenarse. Tampoco define cómo debe presentarse ante el destinatario.

El poder del SMTP es su sencillez y ello se debe a la naturaleza dinámica de las redes a inicios de la década de 1980 (el protocolo SMTP se definió originalmente en 1982). En aquel entonces, los entusiastas conectaban redes entre sí con cualquier clase de goma de mascar y pegamentos. SMTP fue el primer estándar de correo electrónico que fue independiente del mecanismo de transporte. Ello significa que la gente que utilizaba redes TCP/IP podía utilizar el mismo formato para enviar un mensaje que alguien que usara un par de latas y una cuerda.

SMTP también es independiente del sistema operativo, lo que significa que cada sistema puede elegir la forma como almacenará correo electrónico sin preocuparse por la forma como el remitente del mensaje almacena su correo. Puede establecer un paralelo con la forma como operan las compañías telefónicas: cada proveedor de servicios telefónicos tiene su propio sistema de contabilidad independiente. Sin embargo, todos han establecido un estándar para enlazar sus redes de tal manera que las llamadas puedan ir de una red a otra en forma transparente.

Detalles rudimentarios del SMTP

¿Alguna vez ha recibido de un “amigo” un correo electrónico de parte de alguna agencia gubernamental informándole que usted debe una cuantiosa suma de impuestos y multas adicionales? Alguien, en algún lugar, se encarga de que un mensaje como éste sea recibido en el buzón de muchas personas cada vez que se acerca el Día de los Inocentes. Le vamos a mostrar cómo lo hicieron y, lo que es más divertido, cómo puede hacerlo usted mismo (desde luego, tome en cuenta que no estamos promoviendo este tipo de comportamiento).

El propósito de este ejemplo es mostrarle cómo el protocolo SMTP envía un mensaje de un anfitrión a otro. Después de todo, más importante que el aprendizaje sobre la creación de un correo electrónico es el aprendizaje sobre la localización de problemas relacionados con el correo. Así que, en este ejemplo, usted está actuando como el anfitrión que envía, mientras que la máquina con la que se conecta es el anfitrión que recibe.

El protocolo SMTP sólo requiere que un anfitrión pueda enviar texto plano en código ASCII hacia otro anfitrión. Es usual que ello se realice al contactar el puerto SMTP (puerto 25) en un servidor de correo. Puede hacerlo utilizando el programa Telnet. Por ejemplo,

```
[root@servidorA /root]# telnet servidordecorreo 25
```

donde **servidordecorreo** es el servidor del destinatario. El 25 luego de a **servidordecorreo** le dice a Telnet que quiere comunicarse con el puerto 25 del servidor en vez del puerto normal 23 (el puerto 23 se usa para acceso remoto, mientras que el puerto 25 es para el servidor SMTP).

El servidor de correo le responderá con un mensaje de saludo como éste:

```
220 mail ESMTP Postfix
```

Entonces sabrá que está comunicándose directamente con el servidor SMTP.

Aunque hay varios comandos SMTP, los cuatro que vale la pena destacar son

- ▼ **HELO**
- **MAIL FROM:**
- **RCPT TO:**
- ▲ **DATA**

El comando **HELO** se usa cuando un cliente se presenta ante el servidor. El parámetro para **HELO** es el nombre del anfitrión que está generando la conexión. Claro, la mayoría de los servidores de correo toman esta información con un granito de sal y lo verifican. Por ejemplo,

```
HELO cafe-recontra-archi-cargado.com
```

Si la solicitud no viene del dominio cafe-recontra-archi-cargado.com, muchos servidores le responderán que saben su verdadera dirección IP pero no cerrarán la conexión (algunos servidores de correo incluirán un comentario preguntándole por qué usted no utilizó un statement **HELO** veraz).

El comando **MAIL FROM:** requiere la dirección de correo electrónico del remitente como parámetro. Esto le dice al servidor de correo el origen del mensaje. Por ejemplo,

```
MAIL FROM: juan@dominio.com
```

significa que el mensaje es de juan@dominio.com.

El comando **RCPT TO:** también requiere de una dirección de correo electrónico como parámetro; ésta es la del destinatario. Por ejemplo,

```
RCPT TO: gerente@dominio.com
```

que significa que el mensaje está dirigido a gerente@dominio.com.

Ahora que el servidor sabe quiénes son el remitente y el destinatario, necesita saber cuál es el mensaje que mandará. Esto se especifica con el comando **DATA**. Una vez emitido, el servidor esperará el mensaje completo con un encabezado de información relevante seguido por una línea

en blanco, un punto, y luego otra línea en blanco. Retomando el ejemplo, pepito@chistes.com quizás quiera enviar el siguiente mensaje a mama-de-pepito@chistes.com:

DATA

```
354 Enter mail, end with "." on a line by itself
From: Juan <juan@dominio.com>
To: Mi Jefe <gerente@dominio.com>
Subject: a tiempo y dentro de presupuesto.
Date: Jue, 27 Feb 2006 16:06:29 -0600 (GMT)
```

Para su información, el proyecto está en tiempo y dentro del presupuesto establecido!

.

```
250 NAA28719 Message accepted for delivery
```

Y eso es todo. Para cerrar la conexión, utilice el comando **QUIT**.

Ésta es la técnica base utilizada por las aplicaciones que envían correo electrónico; claro que utilizan código en C en vez de usar Telnet, pero el contenido real intercambiado entre el cliente y el servidor es el mismo.

Implicaciones de seguridad

Sendmail, el servidor de correo que la mayoría de los sitios de Internet utilizan, es el mismo paquete que muchas distribuciones de Linux utilizan. Como cualquier otro software, su estructura interna y su diseño son complejos y requieren de una considerable cantidad de cuidado durante su desarrollo. Sin embargo, en años recientes, los desarrolladores de Sendmail han tomado un enfoque bastante paranoico respecto de su diseño que intenta aliviar estas cuestiones. Los desarrolladores de Postfix fueron un paso más adelante y escribieron su software de servidor desde cero, siempre pensando en la seguridad. Lo que distingue a Postfix es que se distribuye de un modo bastante seguro y nos dejan decidir qué tanto terreno cedemos en la configuración que necesitamos para nuestro propio sitio. Esto significa que recae en nosotros la responsabilidad de mantener el software bien configurado para que sea seguro (y, por ende, que no sea vulnerable a ataques).

Estas son algunas consideraciones que debe tener presente cuando se despliega cualquier servidor de correo electrónico:

- ▼ ¿Qué programas activará un correo electrónico cuando se envía al servidor?
- Y esos programas, ¿están diseñados en forma segura?
- Si no pueden hacerse seguros, ¿cómo podemos limitar los daños que pudieran oca-sionar?
- ▲ ¿Qué permisos requieren esos programas para funcionar?

En el caso de Postfix, necesitamos retroceder y examinar su arquitectura.

El servicio de correo tiene tres componentes distintivos. El *mail user agent (MUA)* (*agente del usuario de correo*) es lo que el usuario ve, con lo que interactúa, como los programas Eudora, Outlook y Pine. Un MUA es responsable sólo de leer correo y de permitir al usuario redactar sus mensajes. El *mail transfer agent (MTA)* (*agente de transferencia de correo*) maneja el proceso de llevar

el correo de un sitio a otro; Sendmail y Postfix son MTA. Por último, el *mail delivery agent (MDA)* (*agente de entrega de correo*) es el que toma el mensaje, una vez que se recibe en un sitio, y lo pone en el buzón del usuario apropiado.

Muchos sistemas de correo integran estos componentes. Por ejemplo, Microsoft Exchange Server integra las funcionalidades de un MTA y un MDA en un solo sistema (si considera la interfase Outlook Web Access para el Exchange Server, entonces también es un MUA). Lotus Domino funciona en forma similar. Postfix, por el contrario, trabaja sólo como MTA, delegando la tarea de la entrega local en otro programa externo. Ello permite que cada sistema operativo o cada configuración de un sitio utilice su propia herramienta de personalización, de ser necesario (esto es, permite utilizar un mecanismo especial para almacenamiento de buzones).

En la mayoría de las configuraciones más rigurosas, los sitios prefieren utilizar el programa Procmail para llevar a cabo la entrega de correo (MDA). Esto se debe a su avanzado mecanismo de filtrado así como a un diseño que es seguro desde la base. Muchas de las configuraciones antiguas han mantenido al programa predeterminado **/bin/mail** para realizar la entrega de correo. Las consecuencias de seguridad al utilizar ese programa en particular varían de un sistema operativo a otro.

INSTALACIÓN DEL SERVIDOR POSTFIX

En esta sección abordaremos la instalación del servidor de correo Postfix. Lo escogimos por su facilidad de uso y porque se escribió desde cero para ser más sencillo de utilizar que Sendmail (el autor de Postfix, además, argumenta que la sencillez dio como resultado una seguridad superior). Postfix puede llevar a cabo muchas de las cosas que el programa Sendmail hace, de hecho, el procedimiento de instalación típica de Postfix reemplaza por completo los binarios de Sendmail.

En esta sección instalamos Postfix en una de dos formas: ya sea utilizando el método RPM (recomendado) o desde el código fuente.

Instalación de Postfix mediante RPM

Para instalar Postfix mediante RPM, utilice la herramienta Yum como sigue:

```
[root@hostA /root]# yum install postfix
```

Esto puede tomar a Yum algunos minutos conforme verifica la disponibilidad de actualizaciones y descarga cualquier software requerido para instalar Postfix.

Como Sendmail es el servidor de correo predeterminado que se instala con la mayoría de las distribuciones, necesitará inhabilitarlo utilizando el comando **chkconfig** para luego habilitar Postfix.

```
[root@servidorA /root]# chkconfig --level 35 sendmail off  
[root@servidorA /root]# chkconfig --level 35 postfix on
```

Por último, activaremos el interruptor para iniciar el proceso Postfix. Con una configuración predeterminada, no hará mucho pero confirmaremos si la instalación trabajó como se esperaba.

```
[root@servidorA /root]# /etc/rc.d/init.d/sendmail stop  
[root@servidorA /root]# /etc/rc.d/init.d/postfix start
```

Instalación de Postfix desde el código fuente

Empiece por descargar el código fuente de Postfix desde <http://www.postfix.org>. Al momento de escribir esta obra, la última versión era la postfix-2.2.4.tar.gz. Una vez que lo haya descargado, utilice el comando **tar** para desempacar el contenido.

```
[root@servidorA src]# tar -xvzf postfix-2.2.4.tar.gz
```

Una vez desempacado, cámbiese al directorio **postfix-2.2.4** y ejecute el comando **make** como sigue:

```
[root@servidorA src]# cd postfix-2.2.4  
[root@servidorA postfix-2.2.4]# make
```

El proceso de compilación completo tardará algunos minutos pero debe terminar sin sobresaltos.

SUGERENCIA Si la compilación falla con un error debido a que no se encontró "db.h" o cualquier otra referencia "db", entonces hay indicios de que su sistema no tiene instaladas las herramientas de desarrollador Berkeley DB. Aunque es posible que usted mismo compile dichas herramientas, no es lo más recomendable. Ello se debe a que Postfix fallará si la versión de DB utilizada en Postfix es diferente de la que otras bibliotecas del sistema estén utilizando. Para arreglar esto, instale el paquete db4-devel utilizando Yum como sigue:

```
[root@servidorA postfix-2.2.4]# yum install db4-devel
```

Debido a que Postfix reemplazará el programa Sendmail activo, querrá hacer una copia de seguridad de los binarios Sendmail. Esto puede hacerse de la siguiente manera:

```
[root@servidorA postfix-2.2.4]# mv /usr/sbin/sendmail /usr/sbin/sendmail.OFF  
[root@servidorA postfix-2.2.4]# mv /usr/bin/newaliases /usr/bin/newaliases.OFF  
[root@servidorA postfix-2.2.4]# mv /usr/bin/mailq /usr/bin/mailq.OFF  
[root@servidorA postfix-2.2.4]# chmod 755 /usr/sbin/sendmail.OFF  
[root@servidorA postfix-2.2.4]# chmod 755 /usr/bin/newaliases.OFF  
[root@servidorA postfix-2.2.4]# chmod 755 /usr/bin/mailq.OFF
```

Ahora tenemos que crear un usuario y un grupo bajo el cual Postfix funcionará. Quizá encuentre que algunas distribuciones ya tienen estas cuentas definidas. Si es así, el proceso de alta del usuario causará un error. Ignore el mensaje de error que aparezca.

```
[root@servidorA postfix-2.2.4]# useradd -M -d /no/where -s /no/shell postfix  
[root@servidorA postfix-2.2.4]# groupadd -r postfix  
[root@servidorA postfix-2.2.4]# groupadd -r postdrop
```

Ahora estamos listos para ejecutar **make install** a fin de instalar el software. Postfix incluye un script interactivo que le preguntará algunos valores sobre la ubicación de ciertas cosas. Apéguese a los valores predeterminados en cada pregunta con sólo pulsar **ENTER** en su teclado.

```
[root@servidorA postfix-2.2.4]# make install
```

Con los binarios instalados, es momento para inhabilitar Sendmail desde los scripts de inicio. Podemos lograrlo con el comando **chkconfig** como sigue:

```
[root@servidorA postfix-2.2.4]# chkconfig --level 35 sendmail off
```

La versión fuente de Postfix incluye un buen script de shell que maneja el proceso de inicio y apagado por nosotros. Por el bien de la consistencia, juntémoslo en un script de inicio estándar que puede ser manejado por **chkconfig**. Utilizando las técnicas vistas en el capítulo 6, crearemos un script llamado **/etc/rc.d/init.d/postfix**. Podemos utilizar el siguiente listado de código para el mencionado script **postfix**:

```
#!/bin/sh
# Postfix      Inicia/Detiene el sistema de correo Postfix
#
#chkconfig:    35 99 01
#
. /etc/rc.d/init.d/functions

[ -f /usr/sbin/postfix ] || exit 0

# Revisa cómo nos llamaron.
case "$1" in
    start)
        echo "Iniciando postfix: "
        /usr/sbin/postfix start
        echo "listo"
        touch /var/lock/subsys/postfix
        ;;
    stop)
        echo -n "Deteniendo postfix: "
        /usr/sbin/postfix stop
        echo "listo"
        rm -f /var/lock/subsys/postfix
        ;;
    *)
        echo "Uso: postfix start|stop"
        exit 1
esac
exit 0
```

Con este script en su lugar, verifique dos veces que sus permisos sean los correctos con el comando **chmod**.

```
[root@servidorA postfix-2.2.4]# chmod 755 /etc/rc.d/init.d/postfix
```

Luego utilizamos **chkconfig** para añadirle los niveles de ejecución apropiados para el arranque.

```
[root@servidorA postfix-2.2.4]# chkconfig --add postfix
[root@servidorA postfix-2.2.4]# chkconfig --level 35 postfix on
```

CONFIGURACIÓN DEL SERVIDOR POSTFIX

Después de seguir los pasos anteriores, ha compilado e instalado el sistema de correo Postfix. El script **make install** terminará y le preguntará por cualquier cambio que hubiere salido mal, como olvidarse de añadir el usuario **postfix**. Ahora que ya ha instalado el servidor Postfix, puede cambiar de directorio a **/etc/postfix** y configurar su software.

El archivo de configuración del servidor Postfix es **/etc/postfix/main.cf**. Es obvio por su nombre que éste es el archivo principal de configuración. El otro archivo de configuración que hay que señalar es **master.cf**. Este último es el archivo de configuración de procesos de Postfix que permite modificar cómo funcionan dichos procesos. Este archivo puede ser útil para poner en marcha Postfix en clientes de tal manera que no acepten correo electrónico ni reenvíos hacia un concentrador de correo central. Para obtener más información sobre la forma de hacer esto, vea la documentación en <http://www.postfix.org>. Ahora pasemos al archivo de configuración **main.cf**.

El archivo main.cf

Este archivo es demasiado grande como para listar todas sus opciones en este capítulo. No obstante, abordaremos las más importantes que le permitirán poner en funcionamiento su servidor de correo. Por fortuna, el archivo de configuración está bien documentado y explica en forma clara para qué se utiliza cada opción.

myhostname

Utilizado para especificar el nombre del anfitrión para el cual Postfix estará recibiendo correo electrónico. Nombres típicos que ejemplifican servidores de correo podrían ser

```
myhostname = servidorA.ejemplo.org
```

mydomain

Especifica el dominio de correo que será atendido, tal como hogarnet.com o yahoo.com.

```
mydomain = ejemplo.org
```

myorigin

Todo correo enviado desde este servidor parecerá como si hubiese venido del servidor indicado por este parámetro. Puede ajustarlo a **\$myhostname** o **\$mydomain** como sigue:

```
myorigin = $mydomain
```

Note que puede utilizar el valor de otros parámetros dentro del archivo de configuración colocando el signo \$ antes del nombre de la variable.

mydestination

Lista los dominios que el servidor Postfix tomará como el destino final del correo electrónico entrante. Es típico que este valor se ajuste con el nombre de anfitrión de la caja y el nombre del dominio, aunque puede contener otros nombres tales como:

```
mydestination = $myhostname, localhost.$mydomain, $mydomain,  
mail.$mydomain, www.$mydomain, ftp.$mydomain
```

Si su servidor tiene más de un nombre, por ejemplo, servidorA.ejemplo.org y servidorA. otro-ejemplo.org, querrá asegurarse de que lista ambos en este parámetro.

mail_spool_directory

Puede poner en funcionamiento el servidor Postfix en dos modos para entrega, directo en el buzón del usuario o en un directorio de almacenamiento central. La manera más típica es almacenar el correo en **/var/spool/mail**. La variable se verá como esto en el archivo de configuración:

```
mail_spool_directory = /var/spool/mail
```

El resultado es que el correo se almacenará para cada usuario bajo el directorio **/var/spool/mail** con cada buzón de usuario representado como un archivo. Por ejemplo, todo correo electrónico enviado a yyang@ejemplo.org se almacenará en el archivo **/var/spool/mail/yyang**.

mynetworks

Esta variable es una opción de configuración importante que permite configurar qué servidores podrán retransmitir a través de su servidor Postfix. En general, querrá permitir la retransmisión entre máquinas locales y nada más. De lo contrario, los *spammers* podrán utilizar su sistema para retransmitir sus mensajes. Un valor de ejemplo que puede contener esta variable es

```
mynetworks = 192.168.1.0/24, 127.0.0.0/8
```

Si define este parámetro, anulará el parámetro de **mynetworks_style**. El parámetro **mynetworks_style** le permite especificar cualquiera de las palabras clave **class**, **subnet** o **host**. Estos ajustes le dicen al servidor que confíe en estas redes a las que pertenece.

CUIDADO Si no configura correctamente la variable **\$mynetworks** y los *spammers* comienzan a utilizar su servidor como retransmisor, pronto comenzará a recibir una oleada de mensajes de administradores de correo enfadados informándole de ello. Además, es una forma rápida de hacer que su servidor aparezca en las listas negras de alguno de los servicios de control de *spam* como ORBS o RBL. Una vez que su servidor está en esta lista, muy pocas personas podrán recibir correo electrónico desde su dominio y habrá de pasar por un penoso camino de pruebas rigurosas para que lo saquen de dicha lista. Peor aún, nadie le dirá que usted ha sido incluido en una lista negra.

smtpd_banner

Esta variable le permite regresar una respuesta personalizada cuando un cliente se conecta a su servidor de correo. Es una buena idea configurar este aviso de manera que no revele datos sobre su servidor. Esto dificulta el camino a los *hackers* que están buscando fisuras en su sistema.

```
smtpd_banner = $myhostname ESMTP
```

Hay muchos otros parámetros en el archivo de configuración. Los verá cuando configure las opciones precedentes. Éstas le permitirán ajustar niveles de seguridad y de depuración, si llegara a necesitarlos. Ahora veremos la puesta en marcha del sistema de correo Postfix y algunas acciones de mantenimiento para su servidor.

Revise su configuración

Postfix incluye una útil herramienta para revisar la configuración actual de su servidor que le ayudará a resolver problemas. Utilícela de la siguiente manera

```
[root@servidorA /root]# postfix check
```

El resultado será una lista de errores encontrados en los archivos de configuración o en los permisos de cualquiera de los directorios que necesita el sistema Postfix. Una ejecución rápida en el sistema que nos ha servido de ejemplo muestra lo siguiente:

```
[root@servidorA /root]# postfix check
postfix: fatal: /etc/postfix/main.cf, line 91: missing '=' after attribute name:
"mydomain example.org"
```

Según parece, se trata de un error de dedo en el archivo de configuración. Cuando regrese a corregir un error en el archivo de configuración, asegúrese de leer el mensaje de error con detenimiento y utilice el número de la línea indicada como guía, no espere precisión absoluta. Ello se debe a que un error de dedo en el archivo puede significar que Postfix detectó el error mucho *después* de donde el error realmente ocurrió. En el ejemplo mostrado, un error que cometimos en la línea 76 no se detectó sino hasta la línea 91 debido a la forma como trabaja la máquina de análisis gramatical. Sin embargo, leyendo con cuidado el mensaje de error supimos que el problema era con "mydomain", así que se trató de una búsqueda rápida antes de encontrar la línea correcta.

Pongamos en marcha la herramienta de verificación una vez más.

```
[root@servidorA /root]# postfix check
[root@servidorA /root]#
```

Perfecto. Estamos listos para poner en marcha Postfix.

INICIO DE OPERACIONES DEL SERVIDOR

Iniciar la operación del servidor de correos Postfix es sencillo y claro. Tan sólo añada la opción **start** al comando **postfix**:

```
[root@servidorA /usr/sbin]# postfix start
```

NOTA Recuerde que, si está utilizando Fedora Core y desea utilizar la versión preinstalada, puede especificar `/etc/rc.d/init.d/postfix start` para iniciar Postfix.

Cuando haga cambios a los archivos de configuración, necesita ordenarle a Postfix que vuelva a cargarse para que los cambios surtan efecto. Haga esto enviando el comando `reload` a `postfix`:

```
[root@servidorA /usr/sbin]# postfix reload
```

Revise la cola de correo

En ocasiones las colas de correo en su sistema se llenarán. Esto puede deberse a fallas en la red u otro tipo de fallas atribuibles a otros servidores. Para revisar la cola de envíos en su servidor utilice el siguiente comando:

```
[root@servidorA /root]# mailq
```

Este comando mostrará todos los mensajes que están en la fila de mensajes de Postfix. Éste es el primer paso para que pruebe y revise si el servidor de correo está operando de manera correcta.

Vacíe la cola de correo

De vez en cuando, después de un corte de energía, el correo puede acumularse y tomará varias horas para que se envíen los mensajes. Utilice el comando `postfix flush` para vaciar todos los mensajes que aparezcan listados en la cola por el comando `mailq`.

El comando newaliases

El archivo `/etc/aliases` contiene una lista de alias para correo electrónico. Se usa para crear listas de correo electrónico y alias para usuarios que son válidas dentro de un sitio. Cuando haga cambios en el archivo `/etc/aliases`, necesitará avisar a Postfix acerca de ello mediante el comando `newaliases`. Este comando regenerará la base de datos de Postfix y le informará cuántos nombres se han agregado.

Asegúrese de que todo funciona

Ahora que ya tiene instalado el servidor de correo Postfix, deberá probar y probar una vez más para corroborar que todo trabaja correctamente. El primer paso para hacer esto consiste en utilizar un agente de usuario de correo local como `pine` o `mutt` para enviarse un correo a usted mismo. Si esto funciona, bien. Puede pasar al envío de correo electrónico hacia un sitio remoto utilizando el comando `mailq` para ver cuando salga el mensaje. El paso final es asegurarse de que puede enviar correo electrónico al servidor desde el exterior (esto es, desde Internet). Si puede recibir correo electrónico desde el mundo exterior, su trabajo está terminado.

Registros de correos

En un sistema Fedora Core o Red Hat, los registros de correo se almacenan en forma predeterminada en **/var/log/maillog** como se define en el archivo de configuraciones syslog. Si necesita cambiar éste, puede modificar el archivo de configuración **syslog/etc/syslog.conf** editando la siguiente línea:

```
mail.*          /var/log/maillog
```

La mayoría de los sitios ejecutan sus bitácoras de esta manera; así que, si tiene problemas puede buscar mensajes de error en el archivo **/var/log/maillog**.

Si el correo todavía no funciona

No se preocupe. SMTP no es siempre fácil de poner en funcionamiento. Si todavía tiene problemas, repase con cuidado todos los pasos mientras busca errores. El primer paso es revisar los mensajes en el registro cronológico, pues éstos pueden mostrarle si otros servidores de correo no están funcionando. Si todo parece en orden ahí, verifique las configuraciones DNS. ¿El servidor de correo es capaz de realizar búsquedas de nombres? ¿Puede realizar búsquedas MX? ¿Pueden otras personas realizar búsquedas del nombre de su servidor de correo? El noventa por ciento de la administración de sistemas es localizar problemas en forma apropiada. Otra fuente para ello es mirar lo que otros han hecho. Visite el sitio Web de Postfix en <http://www.postfix.org>, o haga búsquedas de los problemas que está enfrentando en los grupos de noticias con <http://www.google.com>.

RESUMEN

En este capítulo hemos aprendido acerca del funcionamiento básico del SMTP, así como la configuración y la instalación del servidor de correo Postfix. Con esta información usted tiene conocimientos suficientes para configurar y poner en marcha un servidor de correo de uso industrial.

Si busca mayor información sobre Postfix, comience con la documentación en línea en <http://www.postfix.org/documentation.html>. La documentación está bien escrita y es fácil de seguir. Hay más información acerca de la forma como Postfix se puede ampliar para realizar un cierto número de funciones adicionales, pero están fuera del alcance de este capítulo.

Otra excelente referencia sobre el sistema Postfix es *The Book of Postfix: State-of-the-Art Message Transport* por Ralf Hildebrandt y Patrick Koetter (No Starch Press, 2005). Este libro cubre el sistema Postfix con suficiente detalle y es el texto más reciente publicado sobre el tema al momento de escribir esta obra.

Como sucede con cualquier otro servicio, no se olvide que debe mantenerse al tanto de lo que ocurre con Postfix. Las actualizaciones de seguridad se liberan de vez en cuando y es importante que actualice su servidor para incorporar estos cambios.

CAPÍTULO 20



POP e IMAP

En el capítulo 19 cubrimos las diferencias entre los agentes de transferencia de correo (MTA), los agentes de entrega de correo (MDA) y los agentes del usuario de correo (MUA). Cuando se trataba de la entrega de correo a los buzones de usuarios específicos, supusimos el uso de Procmail, el cual entrega copias de correo electrónico a los usuarios en el formato mbox. Este formato de texto plano que pueden leer varios agentes de correo para ventanas de texto como Pine, Elm y Mutt, así como algunos lectores para la interfaz gráfica del usuario.

Sin embargo, el detalle con el formato mbox es que el cliente tiene acceso directo al archivo mbox. Esto no presenta mayor problema en ambientes estrechamente administrados donde el administrador del servidor de correo es también el administrador de los anfitriones cliente; no obstante, este sistema de administración de carpetas de correo se desmorona bajo circunstancias modernas. Cuatro casos específicos muestran con claridad situaciones inoperantes:

- ▼ Los usuarios no pueden estar razonablemente conectados a una red rápida y segura como para acceder al sistema de archivos donde reside el archivo mbox (por ejemplo, los usuarios de computadoras portátiles).
- Los usuarios requieren copias locales de su correo electrónico para examinarlos sin estar conectados.
- Los requerimientos de seguridad dictan que los usuarios no deben tener acceso al almacén de correo (por ejemplo, los directorios base de correo NFS compartidos son inaceptables).
- ▲ Los agentes del usuario de correo no aceptan el formato mbox (sobre todo los clientes basados en Windows).

Para lidiar con estos casos se creó el Post Office Protocol (POP) (Protocolo de oficina de correos) a fin de permitir acceso basado en red a los almacenes de correo. Muchos de los primeros clientes de correo basados en Windows utilizaron el protocolo POP para acceder a correo electrónico en Internet debido a que permitía a los usuarios acceder a los servidores de correo basados en UNIX (los servidores dominantes de correo electrónico en Internet hasta el surgimiento de Microsoft Exchange en la década de 1990).

El concepto en el que se basa el POP es sencillo: un servidor de correo central es administrado de tal forma que siempre permanece en línea y puede recibir correo para todos sus usuarios. El correo recibido hace cola en el servidor hasta que un usuario se conecta mediante el POP y descarga el correo que hace cola, vaciando así su buzón. El correo en el servidor se puede almacenar en cualquier formato (por ejemplo, mbox), en tanto se apegue al protocolo POP. Cuando un usuario quiere enviar correo electrónico, lo envía a través de un servidor central utilizando SMTP. Esto permite al cliente desconectarse de la red dejando que el servidor permanentemente conectado lidie con la tarea de direccionar el mensaje al servidor destino correcto, haciendo cargo de las retransmisiones, retrasos, etc. La figura 20-1 muestra esta relación.

Los primeros usuarios del POP encontraron ciertas limitaciones del protocolo que les parecieron demasiado restrictivas. No existían características tales que permitieran al servidor contar con la capacidad de mantener una copia maestra del correo electrónico del usuario con sólo una copia almacenada en el cliente. Esto llevó a la creación del Interactive Mail Access Protocol (IMAP) (Protocolo de acceso interactivo al correo), cuya primera encarnación de la versión RFC fue IMAP2 en 1988 (RFC 1064). El protocolo IMAP se amplió a la versión 4 (IMAPv4) en 1994. La mayoría de los clientes son compatibles con IMAPv4. Revisiones más recientes lo han llevado a IMAPv4rev1 (RFC 3501).

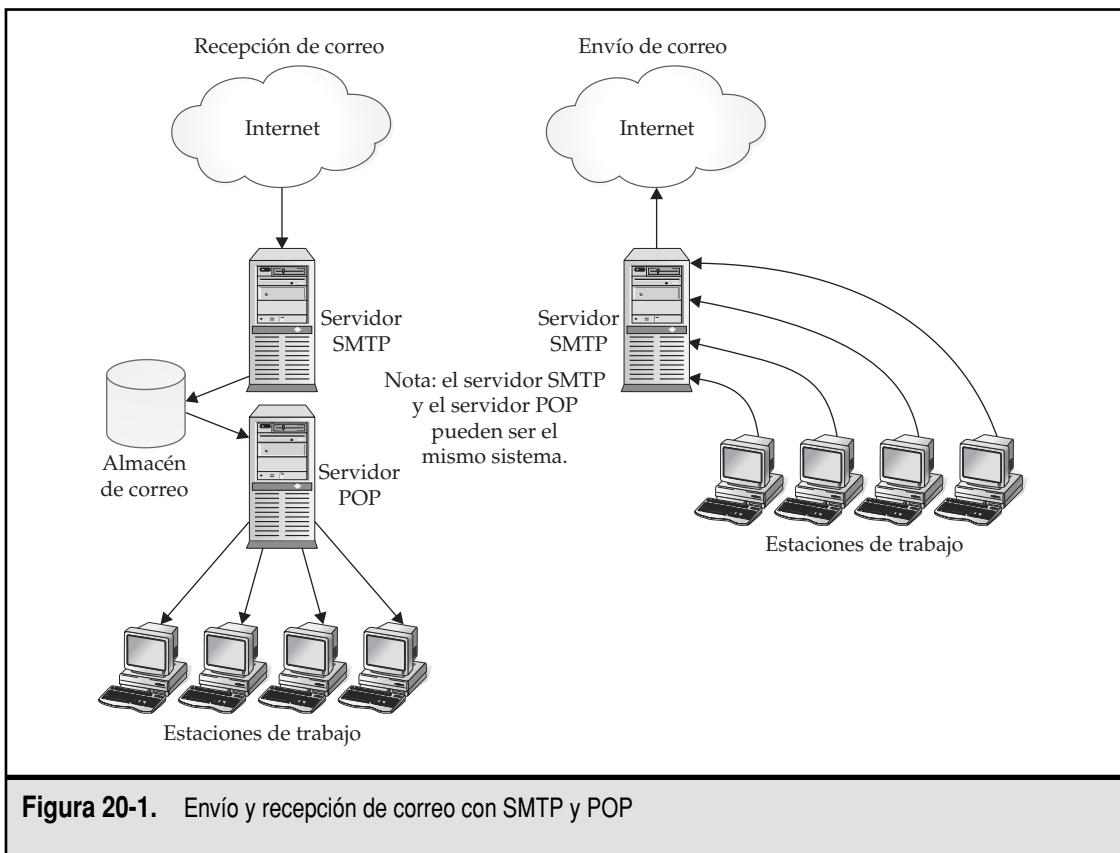


Figura 20-1. Envío y recepción de correo con SMTP y POP

La esencia de la evolución de IMAP se puede entender mejor al pensar que el acceso al correo opera en alguno de tres distintos modos: en línea, fuera de línea y desconectado. El modo *en línea* es semejante a tener acceso directo al sistema de archivos en el almacén de correo (por ejemplo, acceso de lectura al `/var/mail`). El modo *fuerza de línea* es como trabaja POP, donde se supone que el cliente está desconectado de la red excepto cuando está explícitamente descargando su correo electrónico. En este modo, el servidor normalmente no conserva una copia del correo.

El modo *desconectado* trabaja permitiendo a los usuarios conservar copias almacenadas de sus correos. Cuando están conectados, cualquier correo electrónico entrante o saliente es reconocido y sincronizado al instante; sin embargo, cuando el cliente se desconecta, los cambios realizados en el cliente se conservan hasta la reconexión, cuando la sincronización vuelve a ocurrir. Debido a que el cliente sólo mantiene una copia almacenada, un usuario puede cambiarse a un cliente totalmente diferente y vuelve a sincronizar su correo electrónico.

Al utilizar el protocolo IMAP tendrá un servidor de correo que mantendrá los tres modos de acceso.

Después de todo, sostener ambos POP e IMAP es en general una buena idea. Ofrece a los usuarios la libertad de escoger el protocolo y el cliente de correo que mejor les acomode. En este capítulo cubrimos la instalación y la configuración de un servidor UW IMAP, el cual incluye un

enlace a un servidor POP. Este servidor de correo en particular ha estado disponible por varios años y viene con un proceso de instalación fácil de usar. Debe trabajar bien para una base de usuarios pequeña o mediana (hasta unos cuantos cientos de usuarios).

Si está interesado en un servidor de correo IMAP de mayor volumen, considere el servidor IMAP Cyrus o el Courier. Ambos ofrecen opciones de crecimiento impresionantes; sin embargo, es necesario un proceso de instalación más complejo.

FUNCIONAMIENTO DE POP E IMAP

Como los otros servicios que hemos tratado hasta ahora, POP e IMAP necesitan, cada uno, un proceso de servidor para manejar solicitudes. Dichos procesos escuchan en los puertos 110 y 143, respectivamente.

Cada solicitud al, y cada respuesta del servidor está en texto plano ASCII, lo cual significa que, para nosotros, es muy fácil probar la funcionalidad del servidor utilizando Telnet (ello es bastante útil cuando hay usuarios afirmando que “el servidor de correo no sirve”, aunque en realidad lo que ocurre es que no están familiarizados con el sistema). Tal como sucede con un servidor SMTP, los servidores POP e IMAP se pueden controlar con un reducido número de comandos.

Para dar un vistazo a los comandos más comunes, vayamos por el proceso de conectarnos y registrarnos a un servidor POP y a un servidor IMAP. Esta sencilla prueba le permite verificar que el servidor sí funciona y está proporcionando una autenticación válida.

Prueba del servidor POP

Iniciaremos por utilizar Telnet para conectarnos con un servidor POP3. En un intérprete de comandos escriba

```
[root@servidorA /root]# telnet miservidorpop3.midominio.com 110
```

El servidor responde como sigue:

```
+OK ready.
```

El servidor ahora está esperando que usted le proporcione un comando (no se preocupe de que usted no vea ningún mensaje). Inicie por proporcionar su nombre de acceso como sigue.

```
USER sunombredeusuario
```

donde **sunombredeusuario** es, claro, su ID de acceso. El servidor entonces responde con

```
+OK Password required for sunombredeusuario
```

Ahora informe al servidor su contraseña utilizando el comando **PASS**.

```
PASS sucontraseña
```

donde **sucontraseña** es su contraseña. El servidor responde así:

```
+OK sunombredeusuario has X messages (Y octets)
```

donde X y Y serán números verdaderos. X representa el número de mensajes en su buzón, y Y el número de bytes en su buzón. Ahora está registrado y puede emitir comandos para leer su correo. Como nada más estamos verificando que el servidor funciona, podemos terminar la sesión ahora. Utilice el comando **QUIT** y el servidor cerrará la conexión.

Prueba del servidor IMAP

Empezaremos por utilizar Telnet para conectarnos al servidor IMAP. En un intérprete de comandos escriba

```
[root@servidorA /root]# telnet miservidorimap.midominio.com 143
```

El servidor IMAP responde con algo similar a

```
* OK servidorA Cyrus IMAP4 v2.2.6-Invoca-RPM-2.2.6-2.FC3.6 server ready
```

El servidor ahora está listo para recibir sus comandos. Note que, como en el servidor POP, el servidor IMAP no mostrará ningún mensaje.

El formato de comandos con IMAP es

```
<etiqueta> <comando> <parámetro>
```

donde *etiqueta* representa un valor único para identificar (etiquetar) el comando. Los comandos se pueden ejecutar en forma asíncrona, lo que significa que es posible proporcionar un comando y, mientras espera la respuesta, proporcionar otro comando. Como cada comando está etiquetado, la salida mostrará con claridad qué salida corresponde a qué comando.

Para registrarse en el servidor IMAP, utilice el comando **login** como sigue:

```
A001 login nombreusuario contraseña
```

donde **nombreusuario** es el nombre del usuario que deseé probar, y **contraseña** es la que corresponda a ese usuario. Si la autenticación terminó con éxito, el servidor regresará

```
A001 OK User logged in
```

Ello es suficiente para saber dos cosas:

- ▼ El nombre del usuario y la contraseña son válidos.
- ▲ El servidor de correo fue capaz de ubicar y acceder el buzón del usuario.

Una vez validado el servidor, puede terminar la sesión utilizando el comando **logout** como sigue:

```
A002 logout
```

El servidor responderá con algo como esto

```
* BYE LOGOUT received
```

Conocimientos cuando usted necesite ayuda

Steve, el autor de esta obra, en verdad *necesita* leer su correo con cierta regularidad. Nada dejó tan claro este hecho como algo que experimentó años atrás. El servidor en el cual estaba su directorio de inicio estaba caído y no podía registrarse. Éste era un sistema sobre el cual no tenía permisos administrativos, así que tuvo que esperar a que el administrador lo reparara. Desesperado, descargó las especificaciones RFC para POP3 (RFC 1939, <ftp://ftp.isi.edu/in-notes/rfc1939.txt>), desde donde encontró la forma de leer su correo electrónico utilizando comandos POP3 y Telnet. Con paciencia, Steve leyó cada uno de sus correos con este método de acceso (varias docenas de mensajes) y hasta respondió algunos utilizando también Telnet para emitir comandos SMTP directamente al servidor SMTP. Para cuando el administrador del servidor caído regresó, Steve ya había leído y respondido todo su correo electrónico de esa mañana. Increíble, pero cierto.

INSTALACIÓN DEL SERVIDOR UW-IMAP Y POP3

La Universidad de Washington produce un servidor IMAP bastante acreditado que se utiliza en varios sitios de producción alrededor del mundo. Es una implementación bien probada y es la versión de IMAP que instalaremos aquí.

Comience por descargar el servidor UW-IMAP en `/usr/local/src`. Puede encontrar la última versión del servidor en <ftp://ftp.cac.washington.edu/imap/imap.tar.Z>. Una vez descargado, descomprímalo como sigue:

```
[root@servidorA src]# tar -xvzf imap.tar.Z
```

Esto creará un nuevo directorio dentro del cual todo el código fuente estará presente. Para la versión que estamos utilizando, veremos un directorio llamado `imap-2004c1`. Entre a ese directorio como sigue:

```
[root@servidorA src]# cd imap-2004c1
```

La configuración predeterminada que se envía con el servidor UW-IMAP trabaja bien para la mayoría de las instalaciones. Si está interesado en hacer ajustes finos al proceso de construcción, abra con un editor el archivo `Makefile` (que se encuentra en el directorio actual) y léalo. El archivo está documentado al detalle y muestra qué opciones pueden activarse o desactivarse. Para la instalación que estamos haciendo ahora, nos apegaremos a un sencillo cambio en la configuración que podemos hacer en la línea de comandos.

Además de las opciones de construcción, el comando `make` para UW-IMAP requiere que usted especifique el tipo de sistema en el cual el paquete está siendo construido. Esto en contraste con muchos otros programas de fuente abierta que utilizan el programa `./configure` (también conocido como Autoconf) para determinar de manera automática el ambiente en el que funcionan. Las opciones para Linux son las siguientes:

Parámetro	Ambiente
ldb	Debian Linux
lnx	Linux con contraseñas tradicionales
lnp	Linux con Pluggable Authentication Modules (PAM) (Módulos enchufables de autenticación)
lmd	Mandrake Linux (también conocido como Mandriva Linux)
lrh	Red Hat Linux 7.2 y posteriores (cubre Red Hat Enterprise y Fedora Core)
lsu	SuSE Linux
sl4	Linux con contraseñas Shadow (que requieren una biblioteca adicional)
sl5	Linux con contraseñas Shadow (que no requieren una biblioteca adicional)
snx	Linux que requieren de una biblioteca adicional para soporte de contraseñas

¿Algo consternado por las opciones? Ni lo piense. Muchas de ellas son para versiones antiguas de Linux que ya no se utilizan en nuestros días. Si cuenta con una versión de Linux reciente (instalada después de 2002, más o menos), las únicas opciones a las que debe prestar atención son **lsu** (SuSE), **lrh** (Red Hat), **lmd** (Mandrake), **lnp** (PAM) y **ldb** (Debian).

Si está utilizando SuSE, Red Hat/Fedora Core, Debian o Mandrake/Mandiva, utilice la opción apropiada. Si no está seguro, la opción **lnp** debería trabajar en casi todos los sistemas basados en Linux. Es preciso advertir que con la opción **lnp** quizás tenga que editar el archivo Makefile para definir la ubicación de algunas herramientas como OpenSSL (también podría inhabilitar esas características tal como lo haremos en esta instalación).

Para mantener las cosas sin complicaciones, seguiremos un caso genérico y desactivaremos OpenSSL. Para seguir con la construcción, tan sólo ejecute

```
[root@servidorA imap-2004c1]# make lnp SSLTYPE=none
```

El proceso de construcción debería tomar unos cuantos minutos, aun en una máquina lenta. Una vez terminado, tendrá cuatro ejecutables en el directorio: **mtest**, **ipop2d**, **ipop3d** e **imapd**. Copie estos archivos en el directorio **usr/local/bin** como sigue:

```
[root@servidorA imap-2004c1]# cp mtest/mtest /usr/local/bin
[root@servidorA imap-2004c1]# cp ipopd/ipop2d /usr/local/bin
[root@servidorA imap-2004c1]# cp ipopd/ipop3d /usr/local/bin
[root@servidorA imap-2004c1]# cp imapd/imapd /usr/local/bin
```

Asegúrese de que los permisos se ajusten correctamente. Como estos archivos sólo se deben ejecutar por el usuario raíz (root), es recomendable limitar su acceso. Ajuste los permisos como sigue:

```
[root@servidorA imap-2004c1]# cd /usr/local/bin
[root@servidorA bin]# chmod 700 mtest ipop2d ipop3d imapd
[root@servidorA bin]# chown root mtest ipop2d ipop3d imapd
```

Con los archivos necesarios en su lugar, es momento de poner en funcionamiento **xinetd** para que acepte conexiones (para más información sobre **xinetd**, vea el capítulo 8). Agregue las siguientes líneas al archivo **/etc/xinetd.conf**:

```
service imap
{
    flags = REUSE
    socket_type = stream
    wait = no
    user = root
    server = /usr/local/bin/imapd
    log_on_failure += USERID
}

service pop3
{
    flags = REUSE
    socket_type = stream
    wait = no
    user = root
    server = /usr/local/bin/ipop3d
    log_on_failure += USERID
}
```

Antes de ordenar a **xinetd** que vuelva a cargar su configuración querrá verificar que su archivo **/etc/services** tiene ambos pop3 e imap listados. Si **/etc/services** no tiene los protocolos listados, tan sólo añada las siguientes líneas:

```
pop3 110/tcp
imap 143/tcp
```

Por último, ordene a **xinetd** que vuelva a cargar su configuración. Si está utilizando Fedora Core o Red Hat, ello puede llevarse a cabo con el siguiente comando:

```
[root@servidorA bin]# service xinetd reload
```

Si está utilizando otra distribución, encuentre la ID del proceso **xinetd** utilizando el comando **ps** como sigue:

```
[root@servidorA bin]# ps -C xinetd
PID TTY TIME CMD
2013 ? 00:00:00 xinetd
```

Ahora envíe una señal SIGHUP a la ID del proceso **xinetd** utilizando el comando **kill** de la siguiente manera:

```
[root@servidorA bin]# kill -HUP 2013
```

Si todo trabaja como se supone, debería tener en operación un servidor IMAP. Con el uso de los comandos y los métodos antes mostrados en la sección “El funcionamiento de pop e imap”, conéctese al servidor de correo y verifique si un usuario válido puede registrarse en el servidor. Si recibe un mensaje de error durante las pruebas, verifique el archivo `/var/log/messages` para obtener información adicional.

OTROS TEMAS SOBRE SERVICIOS DE CORREO

Aunque ya hay suficiente infraestructura para que pueda empezar con un servidor de correo funcional, existe amplio espacio para mejoras. En esta sección revisaremos algunos de los temas que puede encontrar y los métodos mediante los cuales quizás quiera abordarlos.

Seguridad SSL

El tema de seguridad más relevante con los servidores POP3 e IMAP es que, en su configuración más sencilla, no ofrecen encriptación alguna. Las configuraciones avanzadas del IMAP ofrecen esquemas de pulverización de contraseñas más poderosos; sin embargo, los clientes IMAP comunes como Outlook y Outlook Express, en sus primeras versiones, no los sustentan. Así que su mejor apuesta es la encriptación de todo el torrente de datos utilizando SSL.

A fin de mantener sin complejidades nuestra primera instalación, recordará que no utilizamos SSL en nuestro ejemplo del servidor UW-IMAP. (Siempre es bueno saber que puede poner algo en funcionamiento sin arriesgarse con experimentos innecesarios!) Ahora, si realmente quiere utilizar SSL, necesitará llevar a cabo los siguientes pasos:

1. Recompile UW-IMAP, esta vez con SSL activado.

Cambie el archivo `xinetd.conf` para utilizar los servicios “imaps” y “pop3s” en vez de “imap” y “pop3”, respectivamente (el servicio “imaps” se ejecuta en el puerto TCP 993, y el “pop3s” en el puerto TCP 995).

2. Instale un certificado SSL.

Asegúrese de que su cliente utiliza SSL. En Outlook esta opción es una simple casilla de verificación dentro de la opción “Agregar una nueva cuenta de correo electrónico”.

Compilar de nuevo con SSL activado quizás requiera algo más de experimentación, dependiendo de su instalación. Para los Linux que están definidos (Red Hat/Fedora, SuSE, etc.), las bibliotecas SSL ya están definidas en el archivo Makefile. Si está ejecutando otra distribución, quizás necesite ajustar primero, en forma explícita, las variables SSL en el archivo Makefile.

Por ejemplo, para compilar con la capacidad SSL en Fedora Core 3, ejecute

```
[root@servidorA imap-2004c1]# make clean; make lrh
```

SUGERENCIA Quizás quiera recomendar a todos sus usuarios que utilicen la encriptación SSL para revisar su correo electrónico.

Recuerde copiar los binarios recién compilados al directorio **/usr/local/bin** y ajustar los permisos como se debe.

Respecto de la creación de un certificado SSL, puede crear con bastante facilidad un certificado autofirmado utilizando OpenSSL. Sólo ejecute

```
[root@servidorA imap-2004c1]# openssl req -new -x509 -nodes -out imapd.pem -keyout  
imapd.pem -days 3650
```

Esto creará un certificado con duración de 10 años. Colóquelo en su directorio de certificados OpenSSL. En Red Hat/Fedora, este directorio es **/usr/share/ssl/certs**.

NOTA Si utiliza este método para crear el certificado, los usuarios podrían recibir una advertencia que les informe que dicho certificado no está firmado en forma apropiada. Si no quiere esta advertencia, tendrá que adquirir un certificado de una autoridad competente como VeriSign. Note que estos certificados pueden costar cientos de dólares. Dependiendo de sus usuarios, esto se podría requerir. Sin embargo, si todo lo que necesita es un túnel con encriptación para el paso de contraseñas, un certificado autofirmado funcionará bien.

Pruebas de IMAP con conectividad SSL

Una vez que ponga en marcha un servidor de correo basado en SSL, quizás note que ya no funcionan los trucos para revisar mensajes en el servidor utilizando Telnet. Esto es porque Telnet supone que no hay encriptación en la línea.

Es bastante sencillo esquivar este pequeño obstáculo: utilice OpenSSL como cliente en vez de Telnet, como sigue:

```
[root@servidorA ~]# openssl s_client -connect 127.0.0.1:993
```

En este ejemplo nos podemos conectar con el servidor IMAP que funciona en la dirección 127.0.0.1, aun cuando opere con encriptación. Una vez establecida la conexión, podemos utilizar los comandos vistos en la primera sección de este capítulo.

Disponibilidad

Con la administración de un servidor de correo, pronto aprenderá que el correo electrónico califica como *el más visible* de los recursos en su red. Cuando este servidor se cae, todos se darán cuenta, y lo notarán rápido. De este modo, es importante que considere cómo logrará proporcionar una disponibilidad de 24 horas durante siete días a la semana para los servicios de correo electrónico.

La primera cuestión que amenaza a los servidores de correo es una configuración hecha por “el dedo gordo”. En otras palabras, cometer un error cuando está haciendo administración básica. No hay otra solución a este problema que no sea *¡tener cuidado!* Cuando está lidiando con cualquier servidor de uso industrial, es importante poner atención en cada paso que da y asegurarse de que tenía la intención de hacer lo que está escribiendo. Cuando sea posible, trabaje como un usuario normal en vez de utilizar al usuario raíz y utilice **sudo** para comandos específicos que necesitan permisos raíz.

La segunda cuestión cuando administra servidores de correo es la disponibilidad del hardware. Desgraciadamente, esta amenaza se enfrenta mejor sólo con invertir dinero en una buena máquina, enfriamiento adecuado, y tantas mejoras como su dinero pueda comprar es una buena manera de asegurarse de que su servidor no se caerá por una tontería como una falla en el ventilador de la CPU. Las fuentes de poder duales son otra manera de mantener lejos de usted las fallas mecánicas. Además, las configuraciones de disco en un sistema RAID ayudan a mitigar el riesgo de falla.

Por último, considere la expansión desde la etapa de diseño. Es inevitable que los usuarios consuman todo el espacio disponible en disco. Lo último que quiere hacer es comenzar con malabares de correo electrónico porque el servidor de correo se quedó ¡sin espacio en disco! Para enfrentar esta cuestión utilice volúmenes de disco que puedan ser expandidos en caliente y sistemas RAID que le permitan añadir discos nuevos con rapidez. De esta forma podrá añadir espacio de almacenamiento con un mínimo de tiempos muertos sin tener que mudar operaciones a un nuevo servidor.

Archivos de registro

Aunque ya lo habíamos mencionado en este capítulo, observar los archivos `/var/log/messages` y `/var/log/maillog` es una manera prudente de administrar y seguir la actividad en su servidor de correo. El software de servidor UW-IMAP proporciona una amplia gama de mensajes que le ayudarán a conocer qué está sucediendo con su servidor y también a localizar fallas en cualquier comportamiento particular.

Un excelente ejemplo acerca del uso de archivos de registro surgió cuando escribíamos este capítulo, en específico, la sección SSL. Después de compilar la nueva versión del servidor, olvidamos copiar el archivo `imapd` al directorio `/usr/local/bin`. Esto propició un comportamiento insólito cuando intentamos conectarnos al servidor mediante Outlook. Intentamos hacerlo utilizando el comando `openssl s_client` y nos dio un error irreconocible. ¡¿Qué estaba pasando?!

Un vistazo a los archivos de bitácora utilizando el comando `tail` nos reveló el problema:

```
Apr 27 21:27:37 hostA imapd[3808]: This server does not support SSL
Apr 27 21:28:03 hostA imapd[3812]: imaps SSL service init from 127.0.0.1
```

Bueno, eso más o menos descifra el asunto para nosotros. Repasando nuestros pasos nos dimos cuenta de que olvidamos copiar el nuevo binario `imapd` a `/usr/local/bin`. Una ejecución rápida del comando `cp`, un reinicio de `xinetd`, y de pronto nos estaban dando la bienvenida por nuestro logro.

En resumen, cuando existan dudas, destine unos minutos para revisar los archivos de registro. Quizá encuentre ahí una solución a su problema.

RESUMEN

En este capítulo cubrimos la lógica detrás de IMAP vs. POP3, ejemplos para probar en forma manual la conectividad con estos servidores de correo y el proceso de instalación completo para un servidor UW-IMAP. Con este capítulo ahora tiene suficiente información para hacer funcionar un servidor de correo sencillo capaz de manejar unos cuantos cientos de usuarios sin problema.

Como sucede con cualquier software de servidor que sea visible al mundo exterior, querrá mantenerse al día con las últimas versiones liberadas. Afortunadamente, el paquete que instalamos ha mostrado suficiente estabilidad y seguridad como para mantener al mínimo la necesidad de actualizaciones frecuentes, aunque siempre es recomendable tener un ojo cauteloso.

Por último, abordamos la activación de SSL en su servidor y algunos aspectos básicos para asegurar que su servidor estará disponible las 24 horas durante siete días a la semana. Este método de seguridad es una forma sencilla para evitar que las contraseñas en texto plano incrustadas en el tráfico IMAP caigan en manos que no deberían tenerlas.

Si está interesado en construir un servidor de correo de mayor capacidad, tómese el tiempo para leer sobre los servidores de correo Cyrus y Courier. Ambos proporcionan posibilidades de expansión significativas, aunque requieren de complejidad adicional en su configuración. Sin embargo, si necesita un servidor de correo que atienda una gran cantidad de usuarios, quizás encuentre dicha complejidad adicional como una necesidad.

Para terminar, considere leer los últimos RFC de IMAP y POP a fin de comprender más sobre cada protocolo. Entre más familiarizado esté con los protocolos, le será más sencillo localizar las fallas que se presenten.

CAPÍTULO 21



Secure
Shell (SSH)

Cuando instala su computadora en una red pública (como lo es Internet) observará un efecto secundario adverso: de un momento a otro, algunos de los pillos que pululan por allí querrán irrumpir en su sistema. Como es de suponer, no se trata de algo provechoso.

En el capítulo 15 tratamos varias técnicas para asegurar su sistema Linux, todas ellas enfocadas a limitar al mínimo indispensable el acceso remoto a su sistema. Pero, ¿qué podemos hacer si necesita realizar tareas de administración de sistemas desde un sitio remoto? Telnet es muy inseguro porque transmite la sesión de trabajo, en su totalidad (registro del usuario, contraseña, comandos, todo), en texto plano. ¿Es posible cosechar los beneficios de un sistema verdaderamente multiusuario si no es posible conectarse de forma segura a éste?

NOTA Texto plano significa que los datos no están encriptados. En cualquier sistema, cuando las contraseñas se transmiten en texto plano, una herramienta huelepaquetes podría determinar cuál es la contraseña del usuario. ¡Ello es letal si se trata del usuario raíz!

Para enfrentar este asunto del acceso remoto versus la seguridad de las contraseñas se creó la solución llamada Secure Shell (SSH). Se trata de un conjunto de herramientas de comunicación en red basadas en un protocolo o estándar abierto gobernado por la Internet Engineering Task Force (IETF) (Fuerza de tareas de ingeniería en Internet). Permite a los usuarios conectarse a un servidor remoto tal como lo estarían haciendo con Telnet, **rlogin**, FTP, etc., excepto que la sesión está 100% encriptada. Si alguien utiliza una herramienta huelepaquetes solamente verá tráfico de datos encriptados. Si acaso llegaran a capturar el tráfico encriptado, decodificarlo les tomaría décadas.

En este capítulo daremos un breve y rápido vistazo al concepto de criptografía. Después examinaremos las versiones de SSH, dónde obtenerlo, cómo instalarlo y configurarlo.

CONCEPTOS BÁSICOS SOBRE CRIPTOGRAFÍA DE LLAVE PÚBLICA

Es preciso aclarar lo siguiente antes de que siga leyendo: "Este capítulo, por ningún motivo, pretende ser una autoridad en materia de criptografía y, como tal, no es una fuente definitiva sobre asuntos de criptografía". Lo que encontrará aquí es una discusión general acompañada de algunas referencias a otras obras que abordan el tema con mayor profundidad.

El protocolo Secure Shell se basa en una tecnología llamada *criptografía de llave pública*. Trabaja de manera similar a una caja de seguridad en un banco: se necesitan dos llaves para abrir la caja o, cuando menos, es necesario pasar por varias etapas de seguridad o puntos de revisión. En el caso de la criptografía de llave pública, se requieren dos llaves matemáticas, una pública y otra privada. La llave pública se puede difundir en una página Web de acceso libre, se puede imprimir en una camiseta, o estar expuesta en un pizarrón de avisos en la esquina más transitada de una ciudad. Cualquiera que pida una copia puede recibirla. Por otro lado, la llave privada debe ser protegida bajo las máximas condiciones disponibles. Ésta es la pieza de información que permite que los datos encriptados sean en verdad seguros. Cada combinación de llave pública y llave privada es inimitable.

El proceso de encriptación de datos reales y envío de una persona a otra requiere de varios pasos. Utilizaremos la popular analogía de Alicia y Juan e iremos por el proceso, un paso a la vez, conforme ellos intentan comunicarse entre sí de manera segura. Las figuras 21-1 a 21-5 ilustran una versión muy simplificada del proceso real.

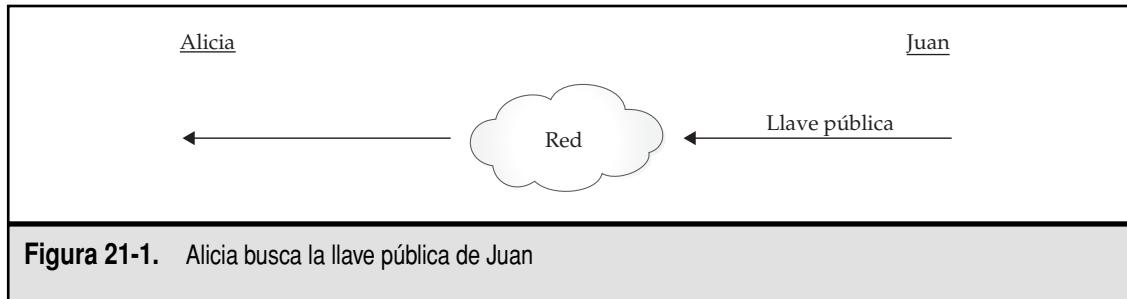


Figura 21-1. Alicia busca la llave pública de Juan

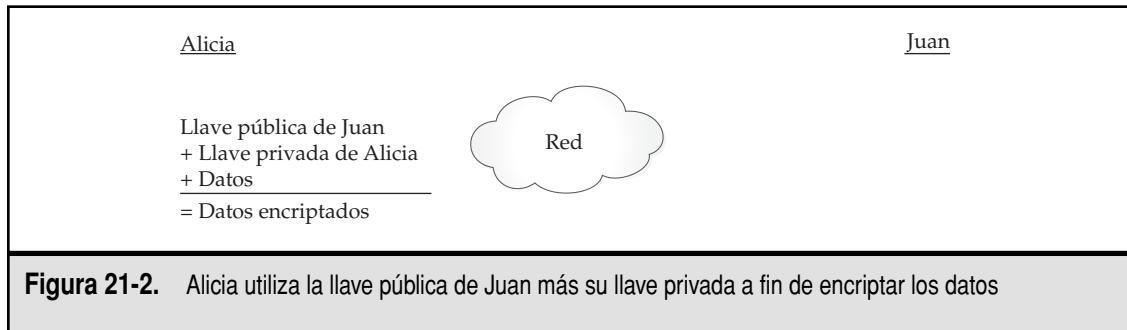


Figura 21-2. Alicia utiliza la llave pública de Juan más su llave privada a fin de encriptar los datos

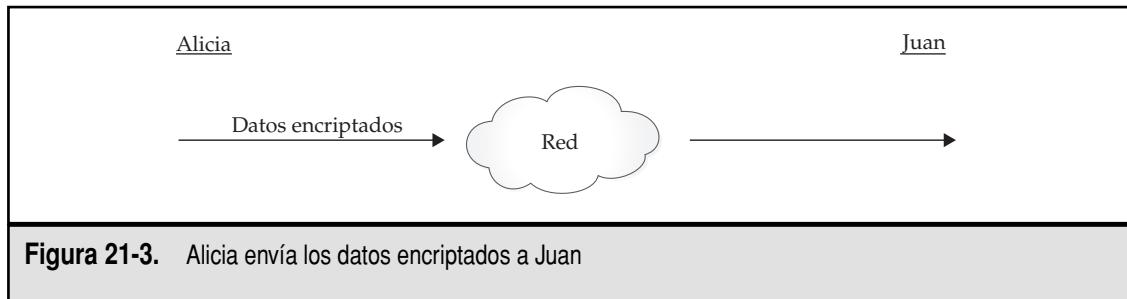


Figura 21-3. Alicia envía los datos encriptados a Juan

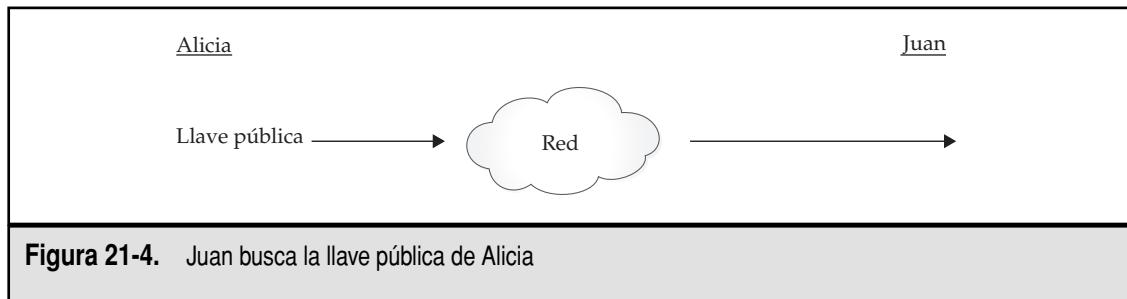
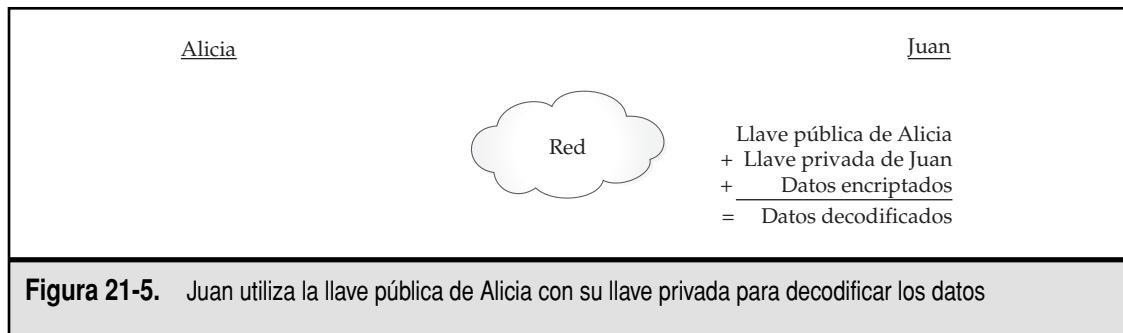


Figura 21-4. Juan busca la llave pública de Alicia



Al ver estos pasos, note que en ningún punto la llave secreta se envía a través de la red. También note que una vez que los datos se encriptan con la llave pública de Juan y firman con la llave privada de Alicia, el único par de llaves que podía decodificarlos eran la llave privada de Juan y la llave pública de Alicia. De modo que, si alguien intercepta los datos durante la transmisión, no podría decodificar los datos sin las llaves privadas correctas.

Para hacer las cosas aún más interesantes, SSH cambia con cierta regularidad su llave de sesión (ésta es una llave simétrica, generada en forma aleatoria, para encriptar la comunicación entre el cliente de SSH y el servidor. Las dos partes la comparten en modo seguro durante la configuración de la conexión de SSH). De esta manera, el torrente de datos queda encriptado en forma distinta cada vez que transcurren algunos minutos. Así, aun cuando alguien pueda descifrar la llave utilizada para la transmisión, ese milagro sólo duraría unos minutos, hasta que las llaves cambien de nuevo.

Características de las llaves

¿Pero qué *es* exactamente una llave? En esencia, una llave es un número muy grande que tiene ciertas propiedades matemáticas. Que alguien pueda romper un esquema de encriptación depende de su habilidad para encontrar cuál es la llave. Así, en tanto más grande sea la llave, más difícil será descubrirla.

La encriptación de bajo grado tiene 56 bits. Ello significa que hay 2^{56} llaves distintas. Para darle idea de la escala, 2^{32} son 4 mil millones, 2^{48} son 256 billones, y 2^{56} son 65 536 billones. Aunque esto parece como un inmenso número de posibilidades, ha sido demostrado que un grupo disperso de PC que se pone a iterar por todas y cada una de las posibilidades puede dar con una llave de encriptación de bajo grado en menos de un mes. En 1998, la Electronic Frontier Foundation (EFF) publicó diseños de una computadora (de aquel entonces) con un costo de 250 000 dólares capaz de romper llaves de 56 bits en unos cuantos segundos y con ello demostrar la necesidad de un grado de encriptación más alto. Si 250 000 dólares le parecen una fuerte suma de dinero, ¡piense en el potencial de fraudes con tarjetas de crédito si alguien utilizará con éxito una computadora para tal propósito!

NOTA La EFF publicó los diseños arriba mencionados en un esfuerzo por convencer al gobierno de Estados Unidos de que las leyes que limitaban la exportación de software criptográfico eran obsoletas y estaban dañando a dicho país debido a que muchas compañías eran forzadas a trabajar en otros países. Al final, esta iniciativa tuvo resultados en 2000, cuando las leyes se suavizaron lo suficiente como para permitir la exportación de software criptográfico de alto grado. En ese momento, para infarto de ese país, la mayoría de las empresas que desarrollan criptografía ya habían mudado sus operaciones de ingeniería hacia otros países.

Los expertos recomiendan que para que una llave sea suficientemente difícil de romper, debe ser igual o mayor que 128 bits. Debido a que cada bit adicional en realidad duplica el número de posibilidades, 128 bits ofrecen un reto genuino. Pero, si en realidad quiere lograr una encriptación sólida, es recomendable utilizar una llave de 512 bits o mayor. SSH puede utilizar hasta 1 024 bits para encriptar datos.

La desventaja de utilizar encriptación de muchos bits es que requiere de mayor poder de procesamiento matemático para que la computadora digiera y valide la llave. Ello requiere tiempo y hace que el proceso de autenticación demore un poco, aunque muchas personas piensan que este retraso bien vale la pena.

NOTA Aunque no está comprobado, se cree que hasta la infame National Security Agency (NSA) de Estados Unidos no puede romper códigos encriptados con llaves de más de 1 024 bits.

Referencias sobre criptografía

El software SSH respalda una variedad de algoritmos de encriptación. Se argumenta que la encriptación de llave pública es el método más interesante para realizar encriptación del tráfico de un sitio a otro y, por varias razones, es también el método más seguro. Si quiere aprender más sobre criptografía, enseguida mostramos algunos buenos libros y otros recursos que puede revisar:

- ▼ *PGP* de Simson Garfinkel et al. (O'Reilly and Associates, 1994)
- *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, segunda edición. Bruce Schneier (John Wiley & Sons, 1995)
- *Cryptography and Network Security: Principles and Practice*, tercera edición. William Stallings (Prentice-Hall, 2002)
- <http://www.ietf.org/internet-drafts/draft-ietf-secsh-25.txt>
- ▲ <http://www.apps.ietf.org/rfc/rfc3766.html>

El libro *PGP* trata en específico sobre el programa PGP, aunque también contiene una cantidad considerable de historia y una excelente colección de textos tutores sobre criptografía en general. El libro *Applied Cryptography* puede ser algo impresionante para muchos, sobre todo para aquellos que no son programadores; sin embargo, explica de manera clara y concisa el funcionamiento de los algoritmos criptográficos (este libro es considerado como una Biblia entre criptógrafos). Por último, *Cryptography and Network Security* se enfoca más en los principios que en la práctica, pero es útil si está interesado en aspectos teóricos de la criptografía en vez del código en sí.

COMPRENSIÓN DE LAS VERSIONES Y DISTRIBUCIONES DE SSH

DataFellows (hoy F-Secure), que publicó la primera versión de SSH, restringía el uso gratuito de SSH a actividades no comerciales; para actividades comerciales era preciso adquirir licencias. Pero más importante que el costo del paquete es el hecho de que el código fuente es completamente abierto. Ello es importante para el software criptográfico debido a que permite a los colegas examinar el código fuente para asegurarse de que no hay agujeros que permitan a los hackers

romper la seguridad (en otras palabras, los criptógrafos serios no se confían de la seguridad obtenida a partir de la oscuridad). Gracias a que el gobierno de EU suavizó algunas de las leyes sobre software de encriptación, el trabajo en el proyecto OpenSSH se ha intensificado, haciéndolo una alternativa viable contra algunas de las versiones comerciales de este protocolo.

Como el protocolo SSH se ha convertido en el estándar IETF, también hay otros desarrolladores trabajando activamente en clientes de SSH para otros sistemas operativos. Hay muchos clientes para la plataforma Windows, para Macintosh, incluso para Palm, además de los de UNIX. Puede encontrar la versión de OpenSSH que abordaremos enseguida en <http://www.openssh.org/>.

OpenSSH y OpenBSD

El proyecto OpenSSH es la punta de lanza del proyecto OpenBSD. Este último es una versión del sistema operativo BSD (otra variante de UNIX) que pugna por la mejor seguridad que se haya podido integrar jamás en un sistema operativo. Una visita rápida a su sitio Web (<http://www.openbsd.org/>) le hará notar que para este software han transcurrido *ocho años* con un solo agujero de seguridad detectado en el acceso remoto que viene en la instalación predeterminada. Por desgracia, este nivel de fanatismo en seguridad tiene una desventaja: deja fuera a las más brillantes, nuevas y completas herramientas que hay disponibles, pues el nuevo inquilino es auditado antes de añadirlo con el fin de que las condiciones de seguridad logradas no disminuyan. Esto ha hecho que OpenBSD sea la plataforma favorita para los firewalls.

Se considera que el núcleo del paquete OpenSSH es parte del proyecto OpenBSD y, por eso mismo, es muy sencillo y específico para el sistema operativo OpenBSD. Para hacer que OpenSSH funcione con otras plataformas, existe un grupo separado que se encarga de crear la edición OpenSSH multiplataforma cuando se liberan nuevas versiones. Es usual que ello ocurra muy rápido, después de la liberación de la original (por ejemplo, OpenSSH 4.1 se liberó el 26 de mayo de 2005 y OpenSSH 4.1p1, la versión multiplataforma, se liberó el mismo día).

NOTA Como estamos enfocados en Linux, utilizaremos la versión que tiene el sufijo *p*, que indica que ya ha sido migrada.

Otros proveedores de clientes de SSH

El cliente de SSH es una parte del paquete del protocolo SSH. Permite a los usuarios interactuar con uno o más servicios proporcionados por el demonio del servidor SSH.

Cada día, muchas personas trabajan con ambientes heterogéneos y es imposible ignorar todos los sistemas Windows 98/NT/2000/XP/2003 y MacOS. A fin de permitir que estas personas trabajen con un sistema operativo *auténtico* (Linux, por supuesto), debe existir un mecanismo para registrarse en forma remota desde tales sistemas operativos. Dado que Telnet no es seguro, SSH proporciona una alternativa. Casi todos los sistemas Linux/UNIX vienen con su propio cliente de SSH interconstruido y, por ende, no hay mucho de que preocuparse; pero es otra historia con los sistemas operativos que no son UNIX. Enseguida mostramos una breve lista de varios clientes de SSH:

- ▼ **PuTTY, para Win32 (<http://www.chiark.greenend.org.uk/~sgtatham/putty>)** Es quizá una de las más viejas y populares implementaciones de SSH para las plataformas Win32. Es de bajo peso: en extremo, un binario sin DLL, sólo el ejecutable. También en este sitio están herramientas como **pscp**, que es una versión de SCP para la línea de comandos de MS Windows.

- **OpenSSH, para MacOS X** En efecto, OpenSSH es parte del sistema MacOS X. Cuando abre la aplicación terminal, con sólo escribir el comando `ssh` lo hará funcionar (también se incluye el servidor SSH OpenSSH).
- **MindTerm (Multiplataforma) (http://www.appgate.com/products/80_MindTerm)** Este programa respalda las versiones 1 y 2 del protocolo SSH. Escrito en Java al 100%, trabaja en muchas plataformas UNIX (incluida Linux) así como Windows y MacOS. Vea la página Web para obtener una lista completa de los sistemas operativos probados.
- **FreeSSH, para Windows (<http://www.freessh.org>)** El sitio Web de FreeSSH trata de seguir todos los programas que implementan el protocolo SSH. El sitio lista implementaciones gratuitas y comerciales de clientes y servidores SSH.
- ▲ **SecureCRT, para Windows (<http://www.vandyke.com/products/securedrt>)** Ésta es una implementación comercial de SSH.

El eslabón más débil

Quizá haya escuchado una variante del viejo refrán que dicta: "La seguridad es tan fuerte como el eslabón más débil". Este refrán en particular tiene una importancia especial en términos de OpenSSH y la seguridad de su red: OpenSSH es tan seguro como lo sea la conexión más débil entre el usuario y el servidor. Esto significa que si un usuario utiliza Telnet para conectarse del anfitrión A al anfitrión B y utiliza `ssh` para conectarse al anfitrión C, toda la conexión puede ser monitoreada interceptando el enlace entre A y B. Entonces, que el enlace entre B y C esté encriptado es ya irrelevante.

Asegúrese de explicar esto a sus usuarios cuando habilite accesos vía SSH, especialmente si está inhabilitando todos los accesos Telnet. Por desgracia, su iniciativa para tapar los huecos de seguridad de esta manera será derrotada con estruendo si sus usuarios utilizan Telnet para conectarse con un anfitrión a través de Internet con el fin de utilizar `ssh` para entrar a su servidor. Y lo más usual será que ni siquiera tengan idea de por qué eso está mal.

NOTA Cuando utiliza Telnet a través de Internet, está cruzando varios límites de red. Cada uno de esos proveedores tiene todo el derecho de "oler" el tráfico y recopilar la información que quiera. Alguien podría verlo a usted leyendo fácilmente su correo electrónico. Con SSH puede confiar en que su conexión es segura.

DESCARGA, COMPILACIÓN E INSTALACIÓN DE SSH DESDE CÓDIGO FUENTE

Como se mencionó anteriormente, casi todas las versiones de Linux se distribuyen con OpenSSH; sin embargo, quizás tenga la necesidad de instalar su propia versión desde el código fuente por la razón que sea (por ejemplo, porque está ejecutando una versión de Linux que se desarrolló ¡en Plutón!). Esta sección cubrirá la descarga del software OpenSSH y los dos componentes que necesita, OpenSSL y zlib. Luego se compilará e instalará el software. Si quiere quedarse con la versión precompilada de OpenSSH que se incluye en su distribución, puede omitir esta sección y pasar directo a la sección que trata sobre la configuración.

Al momento de escribir esta obra, la última versión de OpenSSH es la 4.1p1. Puede descargarla desde <http://www.openssh.com/portable.html>. Seleccione el sitio que sea más cercano a usted y descargue **openssh-4.1p1.tar.gz** a un directorio con suficiente espacio (**/usr/local/src** es una buena elección y es lo que utilizaremos en este ejemplo).

Una vez descargado OpenSSH en **/usr/local/src**, desempáquelo con el comando **tar**, como sigue:

```
[root@servidorA src]# tar xvzf openssh-4.1p1.tar.gz
```

Esto creará un directorio llamado **openssh-4.1p1** dentro de **/usr/local/src**.

Junto con OpenSSH, necesitará OpenSSL versión 0.9.6 o posterior. Al escribir esta obra, la última versión de OpenSSL es **openssl-0.9.8*.tar.gz**. Puede descargarlo desde <http://www.openssl.org/>. Una vez descargado en el directorio **/usr/local/src**, desempáquelo con el comando **tar**, como sigue:

```
[root@servidorA src]# tar xvzf openssl-0.9.8*.tar.gz
```

El último paquete que necesita es la biblioteca zlib, la cual proporciona servicios de compresión y descompresión. Las más modernas distribuciones de Linux ya vienen con este software, pero si quiere la última versión, necesitará descargarla de <http://www.gzip.org/zlib/>. La última versión al momento de escribir esta obra es la versión 1.2.2. Para desempacar el paquete en **/usr/local/src** después de descargarlo, utilice **tar** como sigue:

```
[root@servidorA src]# tar xvzf zlib-1.2.2.tar.gz
```

Los siguientes pasos irán a través del proceso de compilar e instalar todos los componentes de OpenSSH y sus dependientes.

1. Empiece por entrar al directorio **zlib**, como sigue:

```
[root@servidorA src]# cd zlib-1.2.2
```

2. Luego ejecute **configure** y **make**, como sigue:

```
[root@servidorA zlib-1.2.2]# ./configure
```

```
[root@servidorA zlib-1.2.2]# make
```

Esto dará por resultado la biblioteca zlib compilada.

3. Instale la biblioteca zlib al ejecutar lo siguiente:

```
[root@servidorA zlib-1.2.2]# make install
```

La biblioteca resultante será colocada en el directorio **/usr/local/lib**.

4. Ahora necesita compilar OpenSSL. Empiece por cambiarse al directorio donde se desempacó el OpenSSL descargado, como sigue:

```
[root@servidorA ~]# cd /usr/local/src/openssl-0.9.8*
```

5. Una vez que se ha ubicado en el directorio OpenSSL, todo lo que necesita hacer es ejecutar **configure** y **make**. OpenSSL se hará cargo de encontrar en qué plataforma se está ejecutando y se configurará a sí mismo para trabajar de manera óptima. Los comandos exactos son:

```
[root@servidorA openssl-0.9.8*]# ./config  
[root@servidorA openssl-0.9.8*]# make
```

Tome en cuenta que este paso puede tomar algunos minutos para completarse.

6. Una vez que OpenSSL termina de compilarse, puede probarlo ejecutando lo siguiente:

```
[root@servidorA openssl-0.9.8*]# make test
```

7. Si todo salió bien, la prueba debe terminar sin sobresaltos. Si los hubiera, OpenSSL se lo hará saber. Si ello en efecto ocurre, deberá retirar esta copia de OpenSSL e intentar de nuevo el proceso de descarga, compilación e instalación.

8. Una vez terminada la prueba, puede instalar OpenSSL con

```
[root@servidorA openssl-0.9.8*]# make install
```

Este paso instalará OpenSSL en el directorio **/usr/local/ssl**.

9. Ahora está listo para iniciar la compilación e instalación del paquete OpenSSH. Cámbiese al directorio de este paquete, como sigue:

```
[root@servidorA ~]# cd /usr/local/src/openssh-4.1p1
```

10. Como lo hizo con los otros dos paquetes, necesitará empezar ejecutando el programa **configure**. Sin embargo, para este paquete necesitará especificar parámetros adicionales. Es decir, necesitará decirle dónde quedaron instalados los otros dos paquetes. Siempre puede correr **./configure** con la opción **--help** para ver todos los parámetros, pero encontrará que quizás la siguiente instrucción **./configure** trabaje bien:

```
[root@servidorA openssh-4.1p1]# ./configure --with-ssl-dir=/usr/local/ssl/
```

11. Una vez que OpenSSH queda configurado, simplemente ejecute **make** y **make install** para poner todos los archivos en sus directorios **/usr/local** correspondientes.

```
[root@servidorA openssh-4.1p1]# make
```

```
[root@servidorA openssh-4.1p1]# make install
```

Eso es todo. Este conjunto de comandos instalará varios binarios y bibliotecas OpenSSH bajo el directorio **/usr/local**. El servidor SSH, por ejemplo, se colocará en **usr/local/sbin**, y los diversos componentes lo harán bajo el directorio **usr/local/bin**.

INSTALACIÓN DE OpenSSH MEDIANTE RPM

Quizá éste sea el modo más fácil y sencillo de poner en funcionamiento el protocolo SSH en un sistema Linux. Casi podemos asegurar que usted ya tiene el paquete instalado y funcionando si cuenta con una de las más modernas distribuciones de Linux. Inclusive si elige una instalación austera (es decir, la opción con los componentes mínimos) al momento de instalar el sistema operativo, OpenSSH es parte de ese conjunto mínimo de opciones. Esto es más la regla que la excepción. En caso de que usted tuviera una distribución de Linux creada en el planeta Neptuno que al menos tenga RPM instalado, siempre podrá descargar e instalar el paquete RPM precompilado de OpenSSH. En nuestro sistema muestra con Fedora Core puede consultar la base de datos RPM para asegurarse de que OpenSSH ya está instalado escribiendo

```
[root@servidorA ~]# rpm -q openssh  
openssh-4.0p1-3
```

Y si por alguna extraña razón no lo tuviera instalado (o accidentalmente lo desinstaló), puede hacerlo con rapidez utilizando Yum, por medio del siguiente comando:

```
[root@servidorA ~]# yum install openssh
```

En el resto del capítulo se supone que estaremos lidiando con OpenSSH instalado vía RPM.

Inicio y detención del servidor

Si quiere que sus usuarios puedan acceder a su sistema mediante SSH, necesitará asegurarse de que el servicio se esté ejecutando e iniciararlo si no lo está. También deberá asegurarse de que el servicio inicie en forma predeterminada toda vez que reinicie el sistema.

Primero veamos el estatus del demonio **sshd**. Escriba

```
[root@servidorA ~]# service sshd status  
  
sshd (pid 7605 30397 2370) is running...
```

La salida de ejemplo muestra que el servicio se está ejecutando. Pero, en caso contrario, si no lo está, ejecute el siguiente comando para iniciararlo:

```
[root@servidorA ~]# service sshd start
```

SUGERENCIA En un sistema SuSE Linux el comando para verificar el estatus de **sshd** es

```
servidorA:~ # rcsshd status
```

Y para iniciararlo, el comando es

```
servidorA:~ # rcsshd start
```

Si por alguna razón necesita detener el servidor SSH, escriba

```
[root@servidorA ~]# service sshd stop
```

En caso de que realice cambios a la configuración que requiera poner en acción de inmediato, puede reiniciar el demonio en cualquier momento con sólo ejecutar

```
[root@servidorA ~]# service sshd restart
```

El archivo de configuración SSHD

La mayoría de los sistemas Linux, desde el momento en que se instalan, ya tienen configurado y ejecutando el servidor OpenSSH con algunos valores predeterminados. En casi todas las distribuciones Linux basadas en RPM (como Fedora, RHEL o SuSE) es usual que el archivo de configuración **sshd** resida dentro del directorio **/etc/ssh/** y su nombre sea **sshd_config**. Para el OpenSSH que antes instalamos desde el código fuente, el archivo de configuración está ubicado dentro del directorio **/usr/local/etc/**.

Enseguida explicaremos algunas de las opciones de configuración que podrá encontrar dentro del archivo **sshd_config**.

- ▼ **AuthorizedKeysFile** Especifica el archivo que contiene las llaves públicas que pueden ser utilizadas para la autenticación de usuarios. El archivo predeterminado es **/<User_Home_Directory>/.ssh/authorized_keys**.
- **Ciphers** Ésta es una lista separada por comas de cifradores autorizados para la versión 2 del protocolo. Ejemplos de cifradores respaldados son 3des-cbc, aes256-cbc, aes256-ctr, arcfour y blowfish-cbc.
- **HostKey** Define el archivo que contiene una llave privada de anfitrión utilizada por SSH. El archivo predeterminado es **/etc/ssh/ssh_host_rsa_key** o **/etc/ssh/ssh_host_dsa_key** para la versión 2 del protocolo.
- **Port** Especifica el número del puerto en el cual escuchará **sshd**. El valor predeterminado es 22.
- **Protocol** Define las versiones del protocolo a las que **sshd** sostendrá. Los posibles valores son 1 y 2. Tenga presente que el protocolo versión 1 es considerado por lo general como inseguro.
- **AllowTcpForwarding** Especifica si la redirección TCP está permitida. El valor predeterminado es afirmativo (yes).
- ▲ **X11Forwarding** Define si la redirección X11 está permitida. El parámetro debe ser **yes** o **no**. El valor predeterminado es no.

NOTA **sshd_config** es un archivo de configuración algo extraño. Notará que, a diferencia de otros archivos de configuración en Linux, los comentarios (#) denotan los valores predeterminados de las opciones; es decir, los comentarios representan valores predeterminados que ya fueron compilados.

USO DE OpenSSH

OpenSSH tiene varios programas útiles que cubriremos en esta sección. Primero está el programa cliente de **ssh**. Luego está el programa Secure Copy (**scp**). Y, por último, está el programa Secure FTP. La aplicación que probablemente utilizará con mayor frecuencia es el programa cliente de **ssh**.

El cliente de ssh

Con el demonio **ssh** en funcionamiento puede utilizar con sencillez el cliente de **ssh** para conectarse a una máquina desde un sitio remoto de la misma forma en que lo haría si estuviera utilizando Telnet. La principal diferencia entre **ssh** y Telnet es que el primero no le pedirá que se registre, pues supondrá que tiene los mismos datos de acceso en ambas máquinas, como suele suceder.

Sin embargo, si necesita utilizar datos de acceso distintos (por ejemplo, si ya se registró como el usuario raíz en un anfitrión y quiere utilizar **ssh** para conectarse a otro registrándose como usted mismo), todo lo que necesita hacer es proporcionar la opción **-l** junto con los datos de acceso deseados. Por ejemplo, si requiere conectarse al anfitrión servidorB como el usuario yyang desde servidorA, escribiría

```
[root@servidorA ~]# ssh -l yyang servidorB
```

O bien, podría utilizar el comando con el formato **nombreusuario@anfitrion**, como sigue:

```
[root@servidorA ~]# ssh yyang@servidorB
```

El sistema entonces le preguntaría la contraseña del usuario yyang en el servidorB.

Pero si quiere conectarse a un anfitrión remoto sin tener que cambiar los datos de acceso en el lado remoto, tan sólo utilice **ssh** como sigue:

```
[root@servidorA ~]# ssh servidorB
```

Con este comando entrará al servidorB como el usuario raíz.

CREACIÓN DE UN TÚNEL SEGURO

Esta sección cubre lo que algunos conocen como la red privada virtual de los pobres. En esencia, puede utilizar SSH para crear un túnel desde su sistema local hacia un sistema remoto. Esta es una útil característica cuando necesita entrar a una intranet o a un sistema en su intranet que no está expuesto al mundo exterior. Por ejemplo, puede utilizar **ssh** para conectarse a un servidor de archivos que configure la redirección de puertos hacia un servidor Web remoto.

Imaginemos un escenario como el siguiente:

Tenemos un sistema con dos interfaces de red. El sistema tiene como nombre de anfitrión servidorA. Una de las interfaces está conectada en forma directa a Internet. La otra interfaz está conectada a la red de área local (LAN) de una compañía. Suponga que la primera interfaz (la interfaz WAN) tiene una dirección IP tipo pública/ruteable que es 1.1.1.1, mientras que la segunda interfaz tiene una dirección IP tipo privada que es 192.168.1.1. La segunda interfaz está conectada a la LAN (dirección de red 192.168.1.0), la cual está completamente aislada de Internet. El único servicio que está permitido en la interfaz WAN es el demonio **sshd**. La LAN tiene varios servi-

dores y estaciones de trabajo que *sólo* pueden accederse por los anfitriones internos (incluyendo al servidorA).

Suponga que uno de los anfitriones internos es un servidor de aplicaciones de contabilidad basadas en Web al cual el usuario yyang necesita conectarse desde casa. El nombre de anfitrión del servidor Web interno es “contabilidad”, con una dirección IP 192.168.1.100. Y la estación de trabajo que yyang tiene en casa lleva anfitriónA por nombre de anfitrión. Ya dijimos que la red interna está aislada de Internet y los sistemas caseros son parte de la red pública Internet. ¿Cómo se ve este escenario? La figura 21-6 ofrece un diagrama que lo explica todo.

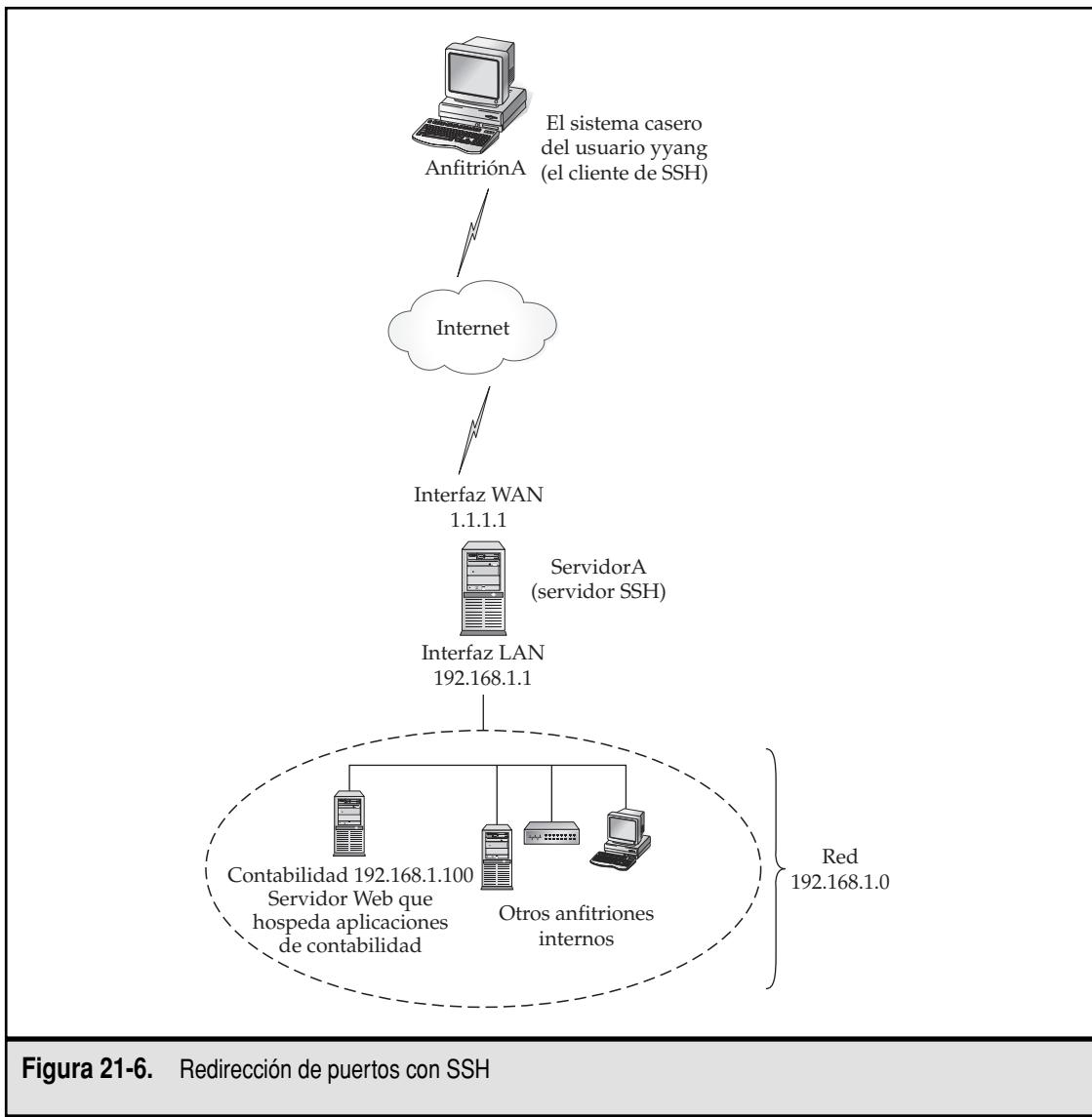


Figura 21-6. Redirección de puertos con SSH

Ahora acceda a la VPN de los pobres, también conocida como túnel SSH. El usuario yyang habilitará un túnel SSH hasta el servidor Web que corre en “contabilidad” con los siguientes pasos:

1. Mientras está sentado frente a su computadora casera, anfitriónA, el usuario yyang se conectará con sus datos de acceso a su sistema local.
2. Una vez dentro, creará un túnel desde el puerto 9000 en el sistema local hacia el puerto 80 en el sistema que corre el software de contabilidad basado en Web, cuyo nombre de anfitrión es “contabilidad”.
3. A fin de lograrlo, yyang se conectará mediante SSH a la interfaz WAN de servidorA (1.1.1.1) utilizando el siguiente comando desde su sistema casero (anfitriónA):

```
[yyang@anfitriónA ~] ssh -L 9000:192.168.1.100:80 1.1.1.1
```

NOTA La sintaxis para el comando de redirección de puertos es:

```
ssh -L puerto_local:anfitrión_destino:puerto_destino servidor_ssh
```

donde **puerto_local** es el puerto local que utilizará para la conexión una vez que el túnel quede listo, **anfitrión_destino:puerto_destino** es la mancuerna anfitrión:puerto hacia la cual el túnel se dirigirá, y **servidor_ssh** es el anfitrión que realizará el trabajo de redirección hacia el anfitrión destino.

4. Después de que yyang se autentifica en forma exitosa a sí mismo en el servidorA y entra a su cuenta en dicho servidor, ahora puede iniciar cualquier navegador Web instalado en su estación de trabajo local (anfitriónA).
5. El usuario yyang necesitará utilizar el navegador Web para acceder al puerto redirigido (9000) en su sistema local a fin de verificar que el túnel está funcionando de manera correcta. Para este ejemplo, es necesario escribir el URL `http://localhost:9000` en el campo donde se escribe la dirección que quiere visitar con el navegador.
6. Si todo sale bien, el contenido Web que se hospeda en el servidor contabilidad debería mostrarse en el navegador Web de la computadora casera de yyang, tal como si estuviera accediendo al sitio desde el interior de la LAN (es decir, dentro de la red 192.168.1.0).
7. Para cerrar el túnel sólo tiene que cerrar todas las ventanas que están utilizándolo y luego cerrar la conexión SSH al servidorA escribiendo `exit` en el intérprete de comandos que utilizó para crear el túnel.

El túnel seguro le permite acceso seguro a otros sistemas dentro de una intranet o desde una ubicación remota. Es una manera grandiosa y de bajo costo para crear una red privada virtual entre un anfitrión y otro. No es una solución de VPN completa debido a que no puede acceder con facilidad a cualquier anfitrión dentro de la red remota, pero sirve para hacer el trabajo. En este proyecto lo que hizo fue redirigir el tráfico HTTP en un puerto. Puede meter al túnel casi cualquier protocolo como VNC o Telnet. Debe notar que esta es una manera para que las personas dentro de un sistema firewall o un proxy salten estos mecanismos y lleguen a computadoras en el mundo exterior.

Trucos del shell de OpenSSH

También es posible establecer un túnel seguro después de haberse registrado en el servidor SSH remoto. Es decir, no tiene que habilitar el túnel cuando está estableciendo la conexión de SSH inicial. Ello es especialmente útil en aquellas ocasiones en las que tiene un shell en un anfitrión remoto y necesita brincar hacia otros sistemas que de otra manera serían inaccesibles.

SSH tiene su propio, particularmente excelente shell, que se puede utilizar para lograr este y otros sensacionales trucos más.

Para obtener acceso al shell que SSH tiene interconstruido, presione SHIFT-~C en el teclado después de haberse conectado a un servidor SSH. En el acto le llegará un mensaje similar al siguiente:

```
ssh>
```

Para habilitar un túnel similar al del ejemplo anterior, escriba este comando en el mensaje/shell **ssh**:

```
ssh> -L 9000:192.168.1.100:80
```

Para salir del shell ssh presione ENTER en su teclado y se irá de regreso a su shell de acceso normal en su sistema.

Si escribe en forma simultánea el carácter tilde (~) y el signo de interrogación (?), podrá ver un listado de todas las cosas que puede hacer en el mensaje **ssh**.

```
[root@serverA ~] # ~?
```

Las siguientes son las secuencias de escape respaldadas:

~.	Termina la conexión
~C	Abre una línea de comandos
~R	Solicita la regeneración de llave (sólo en el protocolo SSH versión 2)
~^Z	Suspende ssh
~#	Lista las conexiones redirigidas
~&	ssh trabaja en el fondo (cuando espera que las conexiones terminen)
~?	Este mensaje
~~	Envía el carácter escape al escribirlo dos veces seguidas

Note que las secuencias de escape sólo se reconocen después de introducir una nueva línea.

Secure Copy (scp)

Secure Copy (**scp**) está pensado como un reemplazo para el comando **rcp**, el cual le permite hacer copias remotas de un anfitrión a otro. El principal problema con el comando **rcp** es que los usuarios tienden a modificar sus configuraciones de modo tal que permiten demasiados accesos al sistema. Para mitigar esto, pida a sus usuarios que utilicen el comando **scp** como sustituto y luego inhabilite por completo el acceso a los programas inseguros **rlogin**. El formato de **scp** es idéntico a **rcp**, de manera que los usuarios no debieran tener problemas con la transición.

Por ejemplo, digamos que el usuario yyang está registrado en su estación de trabajo en casa y quiere copiar un archivo que se llama **.bashrc**, ubicado en su directorio inicial, hacia su directorio inicial en servidorA. El comando para efectuar esto es

```
[yyang@hogarA ~]$ scp .bashrc      servidora:/home/yyang
```

Si quisiera copiar en el sentido opuesto, esto es, desde el sistema remoto, servidorA, hacia el sistema local, hogarA, los parámetros tan sólo necesitan intercambiarse, como sigue:

```
[yyang@anfitriónA ~]$ scp      servidora:/home/yyang/.bashrc ..
```

Secure FTP (sftp)

Secure FTP es un subsistema del demonio **ssh**. Puede acceder al servidor de Secure FTP con la herramienta de línea de comandos **sftp**. Para efectuar un **sftp** como el usuario yyang desde un sistema llamado anfitriónA hacia un servidor de SFTP que se está ejecutando en servidorA, escriba

```
[root@anfitriónA ~]# sftp    yyang@servidorA
```

Enseguida el sistema le preguntará su contraseña tal como sucede cuando utiliza el cliente de **ssh**. Una vez que ha sido autenticado, el sistema mostrará un mensaje para entrada de comandos como la siguiente:

```
sftp>
```

Puede emitir varios comandos SFTP mientras está en este shell. Por ejemplo, para listar todos los archivos y directorios bajo la carpeta **/tmp**, puede utilizar el comando **ls**:

```
sftp> ls -l
drwxr-xr-x    2 yyang      yyang          4096 Jan 30 21:56 Desktop
-rw-r--r--    1 yyang      yyang         1344 Jan 22 21:13 anaconda-huu
.....<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>.....
```

Para listar todos los comandos sólo escriba el signo de interrogación (?). Enseguida se muestran los comandos disponibles con sus comentarios traducidos al español:

```
sftp> ?
Available commands:
cd path           Cambia el directorio remoto a 'path'
lcd path          Cambia el directorio local a 'path'
```

chgrp grp path	Cambia un grupo de archivos de 'path' a 'grp'
chmod mode path	Cambia los permisos del archivo 'path' a 'mode'
chown own path	Cambia el propietario del archivo 'path' a 'own'
help	Muestra este texto de ayuda
get remote-path [local-path]	Descarga el archivo
lls [ls-options [path]]	Muestra el listado del directorio local
ln oldpath newpath	Enlace simbólico al archivo remoto
lmkdir path	Crea un directorio local
lpwd	Imprime el directorio de trabajo local
ls [path]	Muestra un listado del directorio remoto
lumask umask	Ajusta la máscara local para 'umask'
mkdir path	Crea un directorio remoto
put local-path [remote-path]	Sube un archivo
pwd	Muestra el directorio de trabajo remoto
exit	Termina sftp
quit	Termina sftp
rename oldpath newpath	Cambia el nombre de un archivo remoto
rmdir path	Elimina el directorio remoto
rm path	Elimina el archivo remoto
symlink oldpath newpath	Enlace simbólico al archivo remoto
version	Muestra versión de SFTP
!command	Ejecuta el comando 'command' en un shell local
!	Escapa al shell local
?	Sinónimo de ayuda

Notará que los comandos guardan un tremendo parecido a los comandos FTP del capítulo 17. Este cliente es muy útil cuando olvida el nombre completo de un archivo que está buscando.

Archivos utilizados por SSH

Es usual que los archivos de configuración para el cliente y el servidor SSH residan en el directorio **/etc/ssh/** en una distribución (si instaló SSH desde el código fuente en **/usr/local**, la ruta completa será **/usr/local/etc/ssh/**.) Si quiere realizar cambios a los valores predeterminados del cliente de SSH que abarquen todo el sistema, necesita modificar el archivo **ssh_config**.

SUGERENCIA ¡Recuerde que el archivo **sshd_config** es para el demonio del servidor, mientras que el archivo **ssh_config** es para el cliente de ssh!

Dentro del directorio inicial de un usuario la información de SSH se almacena en el directorio **~nombreusuario/.ssh/**. El archivo **known_hosts** se usa para almacenar información sobre la llave del anfitrión. También es utilizado para detener ataques tipo hombre-en-el-medio. SSH lo alertará cuando cambien las llaves del anfitrión. Si las llaves han cambiado por alguna razón válida, como lo sería la reinstalación del servidor, necesitará editar el archivo **known_hosts** y borrar la línea con el servidor que cambió.

RESUMEN

La herramienta Secure Shell es un magnífico reemplazo de Telnet para el acceso remoto. Al adoptar el paquete OpenSSH estará cerrando filas con muchos otros sitios que están inhabilitando por completo el acceso por Telnet y, en vez de ello, están habilitando solamente el acceso de SSH a través de sus firewalls. Dadas las características de amplitud de Internet, este cambio no es algo irrazonable que deba pedir a sus usuarios.

Estos son los principales asuntos que deberá tener presentes cuando considere Secure Shell:

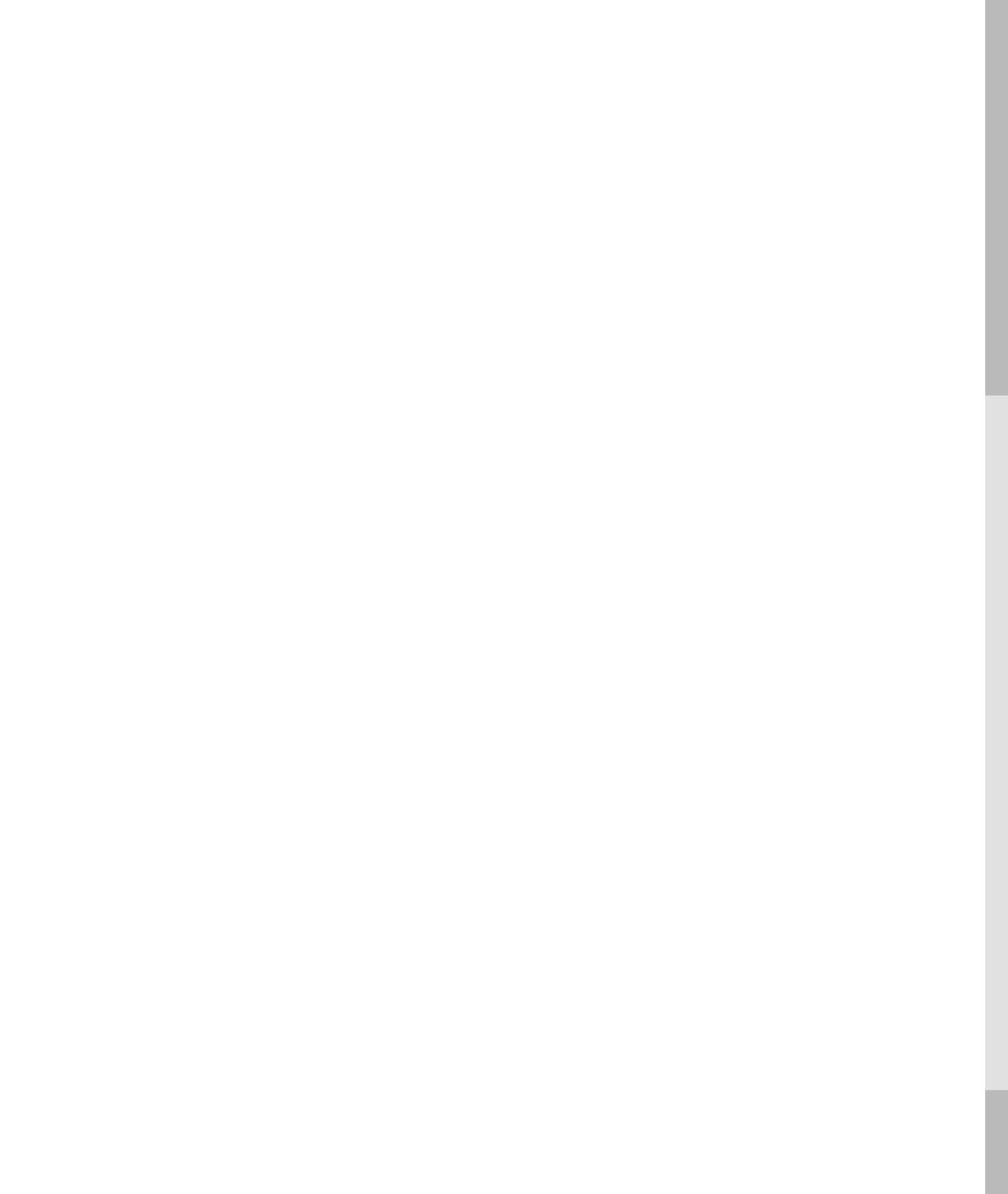
- ▼ SSH es muy fácil de instalar y compilar.
- El reemplazo de Telnet con SSH no requiere de reentrenamiento significativo.
- SSH existe en muchas plataformas, no sólo UNIX.
- ▲ Sin SSH está exponiendo su sistema a ataques de red potencialmente peligrosos en los cuales los criminales pueden “oler” las contraseñas que viajan en las conexiones de Internet.

En suma, debe comprender que el uso de OpenSSH no hace que de inmediato un sistema sea invulnerable. Es decir, no hay forma de sustituir las buenas prácticas de seguridad. Si pone en práctica lo aprendido en el capítulo 15 podrá inhabilitar todos los servicios innecesarios en cualquier sistema que sea expuesto a redes no confiables (como Internet) y permitirá sólo aquellos servicios que son indispensables. Ello significa que, si está ejecutando SSH, debería inhabilitar Telnet, **rlogin** y **rsh**.

PARTE V



Servicios intranet



CAPÍTULO 22



Network File System (NFS)

El Network File System (NFS) (Sistema de archivos en red) es una forma de compartir archivos y aplicaciones en una red, al estilo UNIX. El concepto NFS es algo similar a la forma en que Windows NT comparte discos si consideramos que permite conectarse a uno de ellos y trabajar con éste como si fuera una unidad local, una herramienta útil en verdad para compartir archivos y grandes espacios de almacenamiento entre usuarios.

Pero así como existen similitudes en sus funciones, también hay importantes diferencias en la forma como NFS y MS Windows comparten archivos y aplicaciones, y éstas requieren distintos enfoques en su administración. Asimismo, las herramientas utilizadas para controlar unidades en red son (como es de esperarse) distintas. En este capítulo trataremos esas diferencias; sin embargo, el enfoque principal será en enseñarle a poner en marcha NFS en un ambiente Linux.

FUNCIONAMIENTO DE NFS

Como sucede con la mayoría de los servicios basados en red, NFS sigue el paradigma tradicional cliente-servidor; es decir, tiene componentes del lado del cliente al igual que del lado del servidor.

El capítulo 7 trató sobre los procesos de montar y desmontar sistemas de archivos. La misma idea se aplica a NFS, excepto que ahora cada solicitud de montaje especifica el nombre del servidor que proporciona la unidad compartida. Desde luego, el servidor se debe configurar para permitir que la partición solicitada sea compartida con un cliente.

Veamos un ejemplo. Supongamos que existe un servidor NFS llamado servidorA que necesita compartir su partición local o directorio inicial en una red. En el lenguaje NFS se dice que el servidor NFS exporta su partición /home (inicial). Suponga que hay un sistema cliente en la red llamado clienteA que necesita acceso al contenido de la partición /home exportada por el servidor NFS. Por último, suponga que el resto de los requisitos se cumplen (permisos, seguridad, compatibilidad, etc.).

Para que clienteA pueda acceder al directorio /home en servidorA necesita hacer una solicitud de montaje NFS para que /home sea exportado por servidorA, de manera que clienteA pueda utilizarlo en forma local como el directorio **/home**. El comando para emitir esta solicitud de montaje puede ser tan sencillo como

```
[root@clienteA ~]# mount serverA:/home /home
```

Suponiendo que el comando se ejecutó desde clienteA, todos los usuarios de clienteA podrían ver el contenido de **/home** como si fuera otro directorio. Linux se encargaría de hacer todas las solicitudes al servidor.

Las *Remote procedure calls* (RPC) (*Llamadas a procedimientos remotos*) son responsables de manejar las solicitudes entre el cliente y el servidor. La tecnología RPC proporciona un mecanismo estándar para que el cliente de RPC contacte al servidor y encuentre a cuál servicio se deben dirigir las llamadas. Así, cuando un servicio quiere hacerse disponible a sí mismo en un servidor, necesita autorregistrarse con *portmap*, que administra los servicios RPC. Dicho *portmap* se encarga de decirle al cliente dónde se encuentra el servicio que busca en un servidor.

Versiones de NFS

NFS no es un protocolo estático. Los comités de estándares han ayudado a que NFS evolucione para tomar ventaja de las nuevas tecnologías, así como los cambios en patrones de uso. Al momento de escribir esta obra hay tres versiones bien conocidas del protocolo: NFS versión 2 (NFSv2), NFS versión 3 (NFSv3) y NFS versión 4 (NFSv4). Antes existió NFS versión 1 pero era marcadamente interna para Sun y, como tal, puede decirse que ¡nunca vio la luz del día!

NFSv2 es la más antigua de las tres. NFSv3 es el estándar con mayor aceptación. NFSv4 ha estado en desarrollo por algún tiempo y es el estándar más nuevo. NFSv2 tal vez se debería evitar en la medida de lo posible y sólo debería ser considerado para respaldar el uso de sistemas heredados. Si busca estabilidad y amplio margen de integración en clientes, debería considerar NFSv3. Y sólo si requiere las más recientes características y la compatibilidad hacia atrás no es crucial, entonces debería considerar NFSv4.

Tal vez el factor más importante que debe considerar en la decisión de la versión de NFS que implementará es considerar la versión que sus clientes de NFS mantienen.

Enseguida mostramos algunas de las características de cada versión de NFS:

- ▼ **NFSv2** Las solicitudes de montaje se otorgan en una base unitaria por anfitrión y no por usuario. Utiliza TCP o UDP como su protocolo de transporte. Los clientes de la versión 2 tienen una limitante de 2GB del tamaño de los archivos a los que pueden acceder.
- **NFSv3** Esta versión incluye muchas enmiendas a los errores de programación en NFSv2. Tiene más características que la versión 2 del protocolo. También presenta un mejor desempeño sobre la versión anterior y puede utilizar TCP o UDP como su protocolo de transporte. Dependiendo de los límites del sistema de archivos local en el servidor NFS, los clientes pueden acceder a archivos más grandes que 2GB. Las solicitudes de montaje también se otorgan en una base unitaria por anfitrión y no por usuario.
- ▲ **NFSv4** Esta versión del protocolo utiliza un protocolo como TCP o SCTP como su transporte. Tiene características de seguridad mejoradas gracias a que implementa Kerberos; por ejemplo, la autenticación de clientes puede hacerse en forma unitaria basada en el usuario o con base en una directriz. Se diseñó teniendo a Internet presente y, como resultado, esta versión del protocolo es compatible con firewalls y escucha en el conocido puerto 2049. Los servicios de los protocolos de enlace RPC (por ejemplo, **rpc.mountd**, **rpc.lockd**, **rpc.statd**) no se requieren en esta versión de NFS porque su funcionalidad está interconstruida en el servidor; en otras palabras, NFSv4 combina estos protocolos NFS dispares de antes en una sola especificación de protocolo (el servicio **portmap** tampoco se utiliza). Incluye soporte para atributos de listas para control de acceso a archivos (ACL) y puede mantener clientes de ambas versiones 2 y 3. NFSv4 introduce el concepto del seudosistema de archivos.

La versión de NFS la puede especificar el cliente en el momento del montaje mediante el uso de opciones. Para que un cliente de Linux utilice NFSv2 se usa la opción de montaje **nfsvers=2**. Para NFSv3 se especifica **nfsvers=3**. Pero, en la versión NFSv4, la opción **nfsvers** no se soporta, aunque esta versión se puede usar especificando **nfs4** como el tipo del sistema de archivos.

El resto del capítulo se concentrará sobre todo en NFSv3 porque esta versión es muy estable en Linux, es bastante conocida, y también es la que tiene una amplia gama de plataformas.

Consideraciones de seguridad de NFS

Por desgracia, NFS no es un método muy seguro para compartir discos. Las medidas para hacer que NFS sea más seguro no son muy diferentes de aquellas para hacer más seguro cualquier otro sistema. El único detalle con NFS es que necesita tener plena confianza en los usuarios de un sistema cliente, en especial con el usuario raíz. Si usted es el único que se registra como el usuario raíz en ambos lados (cliente y servidor), entonces hay menos de que preocuparse. En este caso lo importante es asegurarse que los usuarios no son raíz estén imposibilitados para convertirse en usuario raíz, lo cual es algo que de todos modos debería estar haciendo.

Si está en una situación en la cual no puede confiar del todo en la persona con quien necesita compartir el disco, será mejor para su tiempo y esfuerzo que realice la búsqueda de métodos alternativos para compartir el disco (como compartir un disco en modo de sólo lectura).

Como siempre, manténgase al día con los más recientes boletines de seguridad emitidos por el Equipo de Respuesta para Emergencias Computacionales (Computer Emergency Response Team) (<http://www.cert.org/>), y manténgase actualizado con los últimos parches del proveedor de su distribución de Linux.

Montaje y acceso a una partición

Hay varios pasos involucrados con la forma como logra que un cliente haga una solicitud de montaje de una de las particiones de un servidor (estos pasos, en su mayoría, tienen que ver con NFSv2 y NFSv3):

1. El cliente contacta al *portmap* para encontrar cuál puerto de red está asignado al servicio de montajes de NFS.
2. El cliente contacta al servicio de montajes y solicita el uso de una partición. El servicio verifica si el cliente tiene permiso para montar la partición requerida (el permiso que se otorga al cliente se almacena en el archivo */etc/exports*). Si el cliente tiene permiso, el servicio de montaje emite una respuesta afirmativa.
3. El cliente contacta de nuevo al *portmap*, esta vez para saber en qué puerto se localiza el servidor NFS (es usual que éste sea el puerto 2049).
4. Cuando el cliente quiere hacer una solicitud al servidor NFS (por ejemplo, para leer un directorio), una señal RPC se envía al servidor NFS.
5. Cuando finaliza el cliente, actualiza su propia tabla de montajes pero no informa al servidor.

En sentido estricto, la notificación al servidor no es necesaria porque el servidor no lleva control de todos los clientes que han montado su sistema de archivos. Como el servidor no lleva registro sobre el estado de los clientes y éstos no mantienen información sobre el estado del servidor, los clientes y los servidores no pueden diferenciar un sistema caído de uno lento. De esta manera, si reinicia un servidor NFS, todos los clientes retomarán sus operaciones en forma automática tan pronto como el servidor esté de vuelta en línea.

Habilitación de NFS

Casi todas las distribuciones importantes de Linux se envían con respaldo de NFS de una u otra manera. La única tarea que se deja al administrador es configurarlo y habilitarlo. En nuestro sistema de muestra Fedora, habilitar NFS es muy fácil.

Debido a que NFS y sus programas auxiliares se basan en RPC, primero necesita asegurarse de que los servicios del sistema **portmap** están en operación. Para saber el estado del *portmap* en Fedora, escriba

```
[root@servidorA ~]# service portmap status  
portmap is stopped
```

Si el servicio **portmap** no está activo, como lo muestra el resultado de este ejemplo, póngalo en operación como sigue:

```
[root@servidorA ~]# service portmap start  
Starting portmap: [ OK ]
```

Antes de seguir, utilice el comando **rpcinfo** para ver el estado de cualquier servicio basado en RPC que se haya registrado con **portmap**. Escriba

```
[root@servidorA ~]# rpcinfo -p  
program vers proto port  
100000 2 tcp 111 portmapper  
100000 2 udp 111 portmapper
```

Como todavía no hemos puesto en operación al servidor NFS en nuestro sistema muestra, esta salida no presenta muchos servicios de RPC.

Para iniciar el servicio de NFS introduzca este comando:

```
[root@servidorA ~]# service nfs start  
Starting NFS services: [ OK ]  
Starting NFS quotas: [ OK ]  
Starting NFS daemon: [ OK ]  
Starting NFS mountd: [ OK ]
```

Si ahora ejecutamos el comando **rpcinfo** para ver el estado de los programas RPC registrados con el *portmap*, recibiremos la siguiente salida:

```
[root@servidorA ~]# rpcinfo -p  
program vers proto port  
100000 2 tcp 111 portmapper  
100000 2 udp 111 portmapper  
100011 1 udp 952 rquotad  
100003 2 udp 2049 nfs  
....<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>.....  
100003 3 udp 2049 nfs
```

```
100021      1    udp   32791  nlockmgr
100005      1    udp     970  mountd
```

Esta salida muestra que varios programas RPC (**mountd**, **nfs**, **rquotad**, etc.) ahora están en operación.

Para detener al servidor NFS sin tener que apagarlo, introduzca este comando:

```
[root@servidorA ~]# service nfs stop
Shutting down NFS mountd:                                     [  OK  ]
Shutting down NFS daemon:                                    [  OK  ]
Shutting down NFS quotas:                                   [  OK  ]
Shutting down NFS services:                                 [  OK  ]
```

A fin de lograr que el servicio NFS inicie en forma automática la siguiente vez que reinicie el sistema, utilice el comando **chkconfig**. Primero verifique los niveles de ejecución para los cuales NFS está configurado al arranque; escriba

```
[root@servidorA ~]# chkconfig --list nfs
nfs       0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

De manera predeterminada, el servicio está inhabilitado en el sistema Fedora Core; para que se habilite en forma automática escriba

```
[root@servidorA ~]# chkconfig nfs on
```

Los componentes de NFS

Las versiones 2 y 3 del protocolo NFS se apoyan mucho en llamadas a procedimientos remotos (RPC) para manejar las comunicaciones entre clientes y servidores. En Linux, los servicios RPC los administra el servicio **portmap**. Como se mencionó antes, este servicio auxiliar ya no se necesita en NFSv4.

La siguiente lista muestra varios procesos RPC que facilitan el servicio NFS en Linux. Estos procesos son más pertinentes para las versiones 2 y 3, pero se menciona cuando se apliquen para NFSv4.

- ▼ **rpc.statd** Este proceso se encarga de enviar notificaciones a los clientes de NFS cuando se reinicia el servidor NFS. Cuando se le consulta, proporciona información sobre el estado del servidor a **rpc.lockd**. Esto se logra mediante el protocolo Network Status Monitor (NSM) (Monitor de estado de red). Es un servicio opcional que se inicia en forma automática el servicio **nfslock** en un sistema Fedora. No se usa en NFSv4.
- **rpc.rquotad** Como su nombre lo indica, **rpc.quotad** proporciona una interfaz entre NFS y el administrador de cupo. Los clientes/usuarios de NFS estarán limitados por las mismas restricciones de cupo que les aplicarían si estuvieran trabajando en el sistema local de archivos en vez de NFS.

- **rpc.mountd** Cuando se recibe una solicitud para montar una partición, este demonio se encarga de verificar que el cliente cuenta con suficientes permisos para hacer la solicitud. Este permiso se almacena en el archivo **/etc(exports** (en una sección posterior titulada “El archivo de configuración /etc(exports” trataremos a mayor profundidad algunos detalles sobre el asunto). Su inicio se da en forma automática gracias a los scripts **init** del servidor NFS. No se utiliza en NFSv4.
- **rpc.nfsd** Éste es el demonio del servidor NFS y el principal componente del sistema NFS. Trabaja en conjunto con el núcleo de Linux para cargar o descargar el módulo del núcleo conforme sea necesario. Desde luego, todavía es pertinente para NFSv4.

NOTA Debe comprender que NFS en sí es un servicio basado en RPC, sin importar la versión del protocolo. Por lo tanto, aun NFSv4 se basa intrínsecamente en RPC. El detalle está en el hecho de que la mayoría de los anteriores servicios auxiliares basados en RPC (por ejemplo, **mountd**, **statd**) ya no son necesarios porque sus funciones individuales ya se integraron al demonio NFS.

- **rpc.lockd** Este demonio lo usa el demonio **rpc.statd** para manejar la recuperación de candados en sistemas caídos. También permite a los clientes de NFS asegurar archivos en el servidor. Éste es el servicio **nfslock** y ya no se usa en NFSv4.
- **rpc.idmapd** Éste es el demonio de mapeo de nombres de ID en NFSv4. Proporciona esta funcionalidad al núcleo NFSv4 del cliente y del servidor traduciendo identificadores de usuarios y grupos a nombres, y viceversa.
- **rpc.svcgssd** Éste es el demonio **rpcsec_gss** que opera del lado del servidor. El protocolo **rpcsec_gss** permite el uso de las API de seguridad genérica gss-api para proveer seguridad avanzada en NFSv4.
- ▲ **rpc.gssd** Proporciona un mecanismo de transporte del lado del cliente para el mecanismo de autenticación en NFSv4.

Sopporte del núcleo para NFS

Entre las distintas distribuciones de Linux, NFS está implementado de dos formas. La mayoría de las distribuciones se envía con soporte para NFS habilitado en el núcleo. Y unas cuantas distribuciones de LINUX también se envían con soporte para NFS en la forma de demonios independientes que pueden instalarse como un paquete.

Desde que Linux 2.2 se liberó, existe soporte para NFS basado en el núcleo, lo cual hace que funcione mucho más rápido que en implementaciones anteriores. Al momento de escribir esta obra se considera que el soporte para el servidor NFS en el núcleo está listo para irse a producción. No es obligatorio; si no compila este soporte en el núcleo, no lo utilizará. Pero, si tiene la oportunidad de contar con el soporte para NFS desde el núcleo, es altamente recomendable que lo haga. En caso de que no sea así, no se preocupe: el programa **nfsd** que maneja los servicios del servidor NFS es autocontenido en su totalidad y proporciona todo lo necesario para ofrecer servicios NFS.

NOTA En contraparte, los clientes deben tener soporte para NFS desde el núcleo. Este tipo de soporte ha estado presente desde hace mucho tiempo y se sabe que es bastante estable. Casi todas las distribuciones actuales de Linux se envían con soporte habilitado para clientes de NFS desde el núcleo.

CONFIGURACIÓN DE UN SERVIDOR NFS

La puesta en marcha de un servidor NFS es un proceso de dos pasos. El primer paso es crear el archivo **/etc(exports**. Este archivo define cuáles partes del disco del servidor se compartirán con el resto de la red y las reglas mediante las cuales se compartirán (por ejemplo, ¿se le permite a un cliente acceso de sólo lectura al sistema de archivos? ¿Pueden escribir en dicho sistema?). El segundo paso es iniciar el proceso del servidor NFS y que lea el archivo **/etc(exports**.

El archivo de configuración /etc(exports

Este es el archivo de configuración principal del servidor NFS. Este archivo lista las particiones que se pueden compartir, los anfitriones con los que puede compartirse, y los permisos que gobernarán los intercambios. El archivo especifica puntos de montaje remotos para el protocolo de montaje NFS.

El formato para el archivo es muy sencillo. Cada línea en el archivo especifica el punto o los puntos de montaje y las banderas de exportación dentro del sistema de archivos de un servidor local para uno o más anfitriones.

Este es el formato de cada entrada en el archivo **/etc(exports**:

/directorio/a/exportar cliente/direccionip_red(permisos) cliente/direccionip_red(permisos)

- ▼ **/directorio/a/exportar** es el directorio que quiere compartir con otros usuarios, por ejemplo, **/home**.
- **cliente** se refiere al (a los) nombre(s) de anfitrión del (de los) cliente(s) de NFS.
- **direccionip_red** le permite ubicar anfitriones por dirección IP (por ejemplo, 172.16.1.1) o por la dirección IP de la red con combinaciones de máscara (por ejemplo, 172.16.0.0/16).
- ▲ **permisos** son los permisos correspondientes para cada cliente. La tabla 22-1 describe los permisos válidos para cada cliente.

Enseguida mostramos un ejemplo de un archivo **/etc(exports** de NFS completo. Note que la numeración de las líneas se agregó para facilitar las explicaciones que daremos más adelante.

```
1) # Archivo /etc(exports para el servidorA  
2) #  
3) /home      anfitrionA(rw) anfitrionB(rw) clienteA(ro,no_root_squash)  
4) /usr/local  172.16.0.0/16(ro)
```

Las líneas 1 y 2 son comentarios y se ignoran cuando se lee el archivo. La línea 3 exporta el sistema de archivos **/home** a las máquinas llamadas **anfitrionA** y **anfitrionB** y les da permiso de

Opción de permiso	Significado
secure	El número de puerto desde el cual el cliente solicita un montaje debe ser menor a 1024. Este permiso está activado en forma predeterminada. Para desactivarlo, especifique la opción opuesta insecure .
ro	Permite acceso de sólo lectura a la partición. Este es el permiso predeterminado cuando no se especifica nada en forma explícita.
rw	Permite el acceso normal de lectura/escritura.
noaccess	Se negará acceso al cliente a todos aquellos directorios debajo de /dir/a/montar . Esto le permite exportar el directorio /dir al cliente y luego especificar /dir/a como inaccesible sin retirar el acceso a otro directorio como /dir/desde .
root_squash	Este permiso evita que los usuarios raíz en sistemas remotos tengan los privilegios del superusuario raíz local en los volúmenes NFS que montan. El término “squash” empleado aquí significa literalmente <i>aplastar</i> el poder del usuario raíz remoto.
no_root_squash	Permite que el usuario raíz de un cliente de NFS en un anfitrión remoto acceda a un directorio local y lo monte, conservando los mismos derechos y privilegios que el superusuario local tendría normalmente.
all_squash	Redirige todas las UID y GID al usuario anónimo. La opción opuesta es no_all_squash , la cual es la opción predeterminada.

Tabla 22-1. Permisos NFS

lectura/escritura; también lo exporta a la máquina llamada clienteA; le da acceso de sólo lectura pero le permite al usuario remoto raíz obtener privilegios de usuario raíz en el sistema de archivos exportado (/home).

La línea 4 exporta el directorio **/usr/local/** a todos los anfitriones en la red 172.16.0.0/16. Se permite acceso de sólo lectura para los anfitriones dentro de ese rango de red.

Informe al proceso del servidor NFS acerca de /etc(exports)

Una vez que termina de escribir su archivo `/etc(exports)`, utilice el comando `exportfs` para ordenarle a los procesos del servidor NFS que vuelvan a leer la información de configuración. Los parámetros para `exportfs` son los siguientes:

Opción del comando <code>exportfs</code>	Descripción
<code>-a</code>	Exporta todas las entradas en el archivo <code>/etc(exports)</code> . También puede usarse para dejar de compartir los sistemas definidos cuando se utiliza con la opción <code>u</code> , como en <code>exportfs -ua</code> .
<code>-r</code>	Vuelve a exportar todas las entradas en el archivo <code>/etc(exports)</code> . Esto sincroniza <code>/var/lib/nfs/xtab</code> con el contenido del archivo <code>/etc(exports)</code> . Por ejemplo, elimina las entradas de <code>/var/lib/nfs/xtab</code> que ya no están presentes en <code>/etc(exports)</code> y retira las entradas viciadas de la tabla de exportación del núcleo.
<code>-u clienteA:/dir/a/montar</code>	Deja de exportar el directorio <code>/dir/a/montar</code> al anfitrión clienteA.
<code>-o opciones</code>	Las opciones utilizadas aquí son las mismas que las descritas en la tabla 22-1 para los permisos del cliente. Estas opciones se aplicarán sólo al sistema de archivos especificado en la línea de comandos <code>exportfs</code> , no en aquellas especificadas en <code>/etc(exports)</code> .
<code>-v</code>	Activa la mayor cantidad posible de información.

Enseguida mostramos ejemplos de la línea de comando `exportfs`.

Para exportar todos los sistemas de archivos,

```
[root@servidorA ~]# exportfs -a
```

Para exportar el directorio `/usr/local` al anfitrión clienteA con los permisos `read/write` y `no_root_squash`,

```
[root@servidorA ~]# exportfs -o rw,no_root_squash clienteA:/usr/local
```

En casi todos los casos quizás quiera utilizar `exportfs -r`.

Note que los sistemas Fedora Core y RHEL tienen una herramienta GUI bastante capaz (figura 22-1) que se puede usar para crear, modificar y eliminar elementos NFS compartidos. La herramienta se llama `system-config-nfs`. Puede iniciarla desde una línea de comandos escribiendo lo siguiente:

```
[root@servidorA ~]# system-config-nfs
```



Figura 22-1. Utilidad de configuración del servidor NFS

El comando `showmount`

Cuando esté configurando NFS le será de ayuda utilizar el comando `showmount` para verificar que todo esté funcionando como debe ser. Este comando se usa para mostrar información de montajes para un servidor NFS.

Con el uso del comando `showmount` puede determinar con facilidad si ha configurado `nfsd` en forma correcta.

Después de configurar su archivo `/etc(exports` y después de haber exportado todos los sistemas de archivos necesarios mediante el uso de `exportfs`, es momento de utilizar `showmount -e` para ver un listado de todos los sistemas de archivos compartidos en un servidor NFS local. La opción `-e` le dice a `showmount` que muestre la lista de exportaciones del servidor NFS. Por ejemplo,

```
[root@servidorA ~]# showmount -e localhost
Export list for localhost:
/home *
```

El uso del comando **showmount** sin opciones ofrece una lista de los clientes conectados al servidor. Por ejemplo,

```
[root@servidorA ~]# showmount localhost  
Hosts on localhost:  
*  
192.168.1.100
```

También puede utilizar este comando en clientes si proporciona el nombre de anfitrión del servidor NFS como último parámetro. Para mostrar los sistemas de archivo exportados por un servidor NFS (servidorA) desde un cliente NFS (clienteA), puede emitir la siguiente instrucción mientras está registrado en el clienteA:

```
[root@clienteA ~]# showmount -e serverA  
Export list for serverA:  
/home *
```

Localización de fallas del lado del servidor NFS

Cuando exporta sistemas de archivos quizás encuentre en algún momento que el servidor parece rechazar el acceso de clientes, aun cuando el cliente esté listado en el archivo **/etc/exports**. Es usual que esto suceda porque el servidor toma la dirección IP del cliente que se conecta y busca su representación como un nombre de dominio completamente calificado (FQDN), y el anfitrión listado en el archivo **/etc/exports** no está calificado (por ejemplo, el servidor piensa que el nombre del cliente es clienteA.ejemplo.com, pero en el archivo **/etc/exports** aparece registrado sólo como clienteA).

Otro problema común es que el par nombre del anfitrión/dirección IP percibido por el servidor no es el correcto. Esto puede ocurrir debido a un error en el archivo **/etc/hosts** o en las tablas DNS. Necesitará verificar que dicho par sea el correcto.

Para las versiones NFSv2 y NFSv3, el servicio NFS podría fallar en su proceso de arranque si los otros servicios requeridos, como **portmap**, no están activados de antemano.

Aun cuando todo parece marchar en forma correcta del lado del cliente y del lado del servidor, podría ocurrir que el firewall en el servidor impida que el proceso de montaje sea completado. En esas situaciones se dará cuenta de que el comando **mount** parece inmovilizarse sin que muestre errores obvios.

CONFIGURACIÓN DE CLIENTES DE NFS

La configuración de clientes es extremadamente sencilla bajo Linux porque no requiere que se cargue ningún software nuevo o adicional. El único requisito es que el núcleo sea compilado para dar soporte al sistema de archivos NFS. Casi todas las distribuciones de Linux vienen con esta característica habilitada en forma predeterminada. Además del soporte desde el núcleo, el otro factor de importancia es el conjunto de opciones utilizadas con el comando **mount**.

El comando mount

Este comando se trató con anterioridad en el capítulo 7. Los parámetros importantes que debe utilizar con el comando **mount** son: la especificación del nombre del servidor NFS, el punto de montaje local y las opciones especificadas después de **-o** desde la línea de comandos.

Enseguida se presenta un ejemplo de una línea de comando **mount**:

```
[root@clienteA ~]# mount -o rw,bg,soft servidorA:/home /mnt/home
```

Estas opciones del comando **mount** también se pueden usar en el archivo **etc/fstab**. La misma instrucción pero desde el archivo en cuestión se vería así:

```
servidorA:/home      /mnt/home      nfs      rw, bg, soft 0 0
```

En estos ejemplos, **servidorA** es el nombre del servidor NFS. Las opciones **-o** se listan en la tabla 22-2.

Montajes duros vs suaves

En forma predeterminada, las operaciones NFS son *duras*, lo que significa que éstas continúan sus intentos por conectarse al servidor en forma indefinida. Sin embargo, este comportamiento no siempre es conveniente. Causa problemas si un apagado de emergencia ocurre en todos los sistemas. Si sucede que los servidores se apagan antes que lo hagan los clientes, el apagado de éstos se detendrá mientras espera a que los servidores regresen en línea. Por el contrario, si especifica un montaje *suave*, permitirá al cliente cerrar la conexión después de un cierto número de intentos temporizados (especificados con la opción **retrans=r**).

Hay una excepción al uso preferido de un montaje suave con el valor **retrans=r** especificado: no active este comportamiento cuando tenga datos que se deban escribir en el disco en forma obligatoria y usted no quiera controlar el regreso a la aplicación hasta que los datos hayan sido efectivamente escritos (es usual que los directorios de correo montados con NFS se compartan de esta forma).

Montaje cruzado de disco

El *montaje cruzado* es el proceso de tener un servidor A montando discos NFS de un servidor B, y al servidor B montando discos NFS del servidor A. Aunque esto pueda parecer inocuo a simple vista, hay un sutil peligro al hacerlo. Si los dos servidores se caen, y ambos requieren montar discos del otro para completar su secuencia de arranque, entonces tiene en sus manos el típico problema del huevo y la gallina. El servidor A no iniciará hasta que el servidor B termine de iniciar, pero el servidor B no terminará de iniciar porque está esperando que el servidor A termine de iniciar.

Para evitar esta paradoja, asegúrese de no caer en esta situación. Todos sus servidores deben ser capaces de iniciar por completo sin necesidad de montar los discos de nadie más, para nada. Sin embargo, ello no significa en absoluto que no pueda hacer montaje cruzado. Hay razones legítimas para necesitar montaje cruzado, como la facultad de poner a disposición de todos los servidores los directorios de inicio de cada uno.

Opción de comando <code>mount -o</code>	Descripción
bg	Montaje en segundo plano. En caso de que el montaje falle al inicio (por ejemplo, si el servidor falla), el proceso de montaje pasará a procesamiento en segundo plano y continuará intentando su ejecución hasta que lo consiga. Esto es útil para sistemas de archivos montados durante el arranque, pues evita que el sistema se inmovilice con el comando <code>mount</code> si el servidor está caído.
intr	Especifica si el montaje se puede interrumpir. Si un proceso tiene pendientes entradas/salidas (I/O) de datos en una partición montada, esta opción permite que el proceso sea interrumpido y que la llamada de I/O sea descartada. Para más información, vea "Importancia de la opción <code>intr</code> " más adelante en esta sección.
hard	Ésta es una opción predeterminada en forma implícita. Si una operación NFS con archivos excede el temporizador, entonces en la consola se envía el mensaje "server not responding" (el servidor no responde) y el cliente prosigue reintentando en forma indefinida.
soft	Habilita un montaje suave en una partición y permite que el cliente cierre la conexión después de cierto número de reintentos temporizados (especificados con la opción <code>retrans=r</code>). Para más información, vea "Montajes duros vs suaves", en la página anterior.
retrans=n	El valor <i>n</i> especifica el número máximo de reintentos de conexión para un sistema montado en forma suave.
rsize=n	El valor <i>n</i> es el número de bytes que NFS utiliza cuando lee archivos desde un servidor NFS. El valor predeterminado depende del núcleo, pero actualmente es 4096 para la versión NFSv4. El desempeño se puede incrementar en amplia medida al solicitar un valor más alto (por ejemplo, <code>rsize=32768</code>).

Tabla 22-2. Opciones de montaje para NFS

Opción de comando <code>mount -o</code>	Descripción
<code>wsize=n</code>	El valor <code>n</code> especifica el número de bytes que NFS utiliza cuando escribe archivos desde un servidor NFS. El valor predeterminado depende del núcleo pero actualmente es algo así como 4096 para la versión NFSv4. El desempeño se puede incrementar en amplia medida al solicitar un valor más alto (por ejemplo, <code>wsize=32768</code>). Este valor se negocia con el servidor.
<code>proto=n</code>	El valor <code>n</code> especifica el protocolo de red que se va a utilizar para montar el sistema de archivos NFS. El valor predeterminado en las versiones NFSv2 y NFSv3 es UDP. Es usual que los servidores NFSv4 ofrezcan soporte sólo para TCP. Por lo tanto, los protocolos de red válidos son UDP y TCP.
<code>nfsvers=n</code>	Permite el uso de un número de versión RPC alternativo a fin de contactar al demonio NFS en el anfitrión remoto. El valor predeterminado depende del núcleo, pero los valores posibles son 2 y 3. Esta opción no se reconoce en NFSv4, en cuyo caso sólo sería necesario especificar <code>nfs4</code> como el tipo del sistema de archivos.
<code>sec=value</code>	Ajusta el modo de seguridad para las operaciones de montaje a <code>valor</code> . <ul style="list-style-type: none">▼ <code>sec=sys</code> Utiliza UID y GID locales de UNIX para autenticar operaciones de NFS (AUTH_SYS). Este es el valor predeterminado.■ <code>sec=krb5</code> Kerberos V5 en vez de UID y GID locales para autenticar usuarios.■ <code>sec=krb5i</code> Utiliza Kerberos V5 para autenticar usuarios y lleva a cabo verificaciones de integridad de las operaciones NFS mediante el uso de sumas seguras que previenen la alteración de datos.▲ <code>sec=krb5p</code> Utiliza Kerberos V5 para autenticar usuarios y verificar integridad, además de llevar a cabo la encriptación del tráfico NFS para prevenir que sea “olfateado”.

Tabla 22-2. Opciones de montaje para NFS (*cont.*)

En estas situaciones asegúrese de que configura las entradas `/etc/fstab` de modo que utilicen la opción montada `bg`. Al hacerlo, permitirá en cada servidor que el proceso `mount` pase a segundo plano en caso de que falle cualquier montaje; así, todos los servidores tienen la oportunidad de reiniciar por completo para luego ofrecer como se debe las particiones montables con NFS.

Importancia de la opción `intr`

Cuando un proceso hace una llamada al sistema, el núcleo entra en acción. Mientras el núcleo atiende dicha llamada al sistema, el proceso no tiene control sobre sí mismo. En caso de que ocurra un error de acceso en el núcleo, el proceso debe continuar y esperar hasta que el núcleo regrese la solicitud; mientras tanto, el proceso no puede rendirse y abandonar la solicitud. En casos normales, el control del núcleo no es un problema porque es usual que las solicitudes al núcleo se atiendan con celeridad. Sin embargo, cuando se presenta un error, puede convertirse en una verdadera molestia. Debido a ello, NFS tiene la opción de montar particiones con una bandera de interrupción (la opción `intr`), la cual permite que un proceso que está esperando una solicitud NFS realizada pueda interrumpir la espera y seguir con otra cosa.

En general, a menos que tenga una razón específica para no utilizar la opción `intr`, es una buena idea utilizarla.

SUGERENCIA ¡Recuerde mantener las UID en sincronía! Cada solicitud de cliente de NFS incluye la UID del usuario que hizo la solicitud. Esta UID la usa el servidor para verificar que el usuario tenga los permisos de acceso al archivo requerido. Sin embargo, a fin de que la verificación de permisos de NFS funcione de manera correcta, las UID de los usuarios deben sincronizarse entre el cliente y el servidor (la opción `all_squash /etc(exports evita este problema). No es suficiente con tener al mismo usuario registrado en ambos sistemas. Una base de datos NIS o LDAP puede ayudarle en esta situación.`

Ajustes finos al desempeño

El tamaño predeterminado del bloque que se transmite con las versiones 2 y 3 de NFS es de 1KB, mientras que para NFSv4 es de 4KB. Esta medida es ventajosa, pues se ajusta muy bien al tamaño de un paquete y, en caso de que se pierda alguno de ellos, NFS tiene que retransmitir sólo unos cuantos. La desventaja es que no se explotan ciertas condiciones disponibles: la mayoría de las redes son rápidas en grado suficiente como para mantenerse a la par con la segmentación que requiere el transporte de bloques de datos más grandes, además de que la mayoría de las pilas de redes son suficientemente confiables como para que sea extraño perder un bloque de datos.

Con estas características, es una buena costumbre optimizar para el caso de una pila de red veloz y confiable, debido a que ello será lo que tenga disponible 99% de las veces. La manera más fácil de hacerlo con NFS es utilizar las opciones `wszie` (tamaños de escritura) y `rszie` (tamaño de lectura). Un buen tamaño para utilizar es 8KB para las versiones 2 y 3 de NFS. Ese valor es especialmente bueno, sobre todo si tiene tarjetas de red que respaldan marcos grandes.

Un ejemplo de entradas con `wszie` y `rszie` es como sigue:

```
servidorA:/home  /mnt/home  nfs      nfsvers=3,rw,bsize=8192,rszie=8192 0 0
```

LOCALIZACIÓN DE FALLAS DEL LADO DEL CLIENTE DE NFS

Como sucede con cualquier servicio importante, NFS tiene mecanismos que ayudan a lidiar con situaciones de error. En esta sección discutimos algunos casos de errores comunes y la manera como NFS los enfrenta.

Identificadores de archivo viciados

Si un archivo o directorio los usa un proceso cuando otro proceso los elimina, el primer proceso recibe un mensaje de error del servidor. Es común que este mensaje de error sea "Stale NFS file handle" (identificador de archivo NFS viciado).

La mayoría de las veces los identificadores de archivo viciados ocurren cuando está utilizando un equipo con el ambiente gráfico X Window System y tiene dos ventanas de terminales abiertas. Supongamos que la primera ventana está ubicada en un directorio (digamos que es `/mnt/usr/local/mydir`) y ese directorio se elimina desde la segunda ventana. La siguiente vez que oprima ENTER en la primera ventana, entonces verá el mensaje de error.

Para remediar este problema sólo tiene que cambiarse a un directorio que sepa que existe, sin utilizar directorios relacionados (por ejemplo, `cd /tmp`).

Permisos denegados

Es probable que vea el mensaje "Permission denied" (permiso denegado) si está registrado en forma local como el usuario raíz y está intentando acceder a un archivo que está en un sistema de archivos montado mediante NFS. Este mensaje por lo común quiere decir que el servidor donde el sistema de archivos está montado no reconoce los permisos del usuario raíz.

Es habitual que esto sea el resultado de olvidarse que el archivo `/etc(exports)`, en forma pre-determinada, habilitará la opción `root_squash`. Y, de este modo, si está conectándose desde un cliente de NFS con acceso permitido como el usuario raíz, seguro estará preguntándose por qué está recibiendo errores de permiso denegado aun cuando el NFS remoto parece montado correctamente.

La manera más rápida de esquivar este problema es convertirse en el usuario propietario del archivo que está intentado controlar. Por ejemplo, si usted es raíz y está intentando acceder a un archivo propiedad del usuario pp_propietario, utilice el comando `su` para convertirse en pp_propietario:

```
[root@clienteA ~]# su - pp_propietario
```

Cuando termine de trabajar con el archivo, puede salir del intérprete de comandos de pp_propietario y regresar a raíz. Note que con este método se supone que pp_propietario existe como un usuario en el sistema y que tiene la misma UID tanto en el cliente como en el servidor.

Un problema similar ocurre cuando usuarios que tienen el mismo nombre en el cliente y en el servidor de todas maneras obtienen errores de permisos denegados. Esto quizás se deba a que las UID asociadas al nombre del usuario son distintas entre los sistemas. Por ejemplo, pp_propietario podría tener 501 como UID en el anfitrión clienteA mientras que el mismo pp_propietario podría tener 600 como UID en el servidorA. La solución más sencilla a este problema sería crear usuarios con las mismas UID y GID en todos los sistemas. La solución escalable a este problema podría ser la implementación de infraestructura a fin de mantener una base de datos para administración centralizada de usuarios como LDAP o NIS, de modo que todos los usuarios tuvieran las mismas UID y GID independientemente de sus sistemas locales de clientes.

CONFIGURACIÓN DEL CLIENTE Y DEL SERVIDOR NFS

En esta sección reuniremos todo lo que hemos aprendido hasta ahora caminando paso a paso a través del establecimiento real de un ambiente NFS. Realizaremos la configuración y puesta en marcha del servidor NFS. Despues pondremos en marcha un cliente NFS y nos aseguraremos de que los directorios sean montados cuando el sistema arranca.

En particular, exportaremos el sistema de archivos **/usr/local** en el anfitrión servidorA hacia un anfitrión particular en la red llamado **clienteA**. Queremos que clienteA tenga permisos de lectura y escritura en el volumen compartido y que el resto del mundo tenga acceso de sólo lectura a este mismo recurso. Nuestro clienteA montará el recurso NFS en el punto de montaje **/mnt/usr/local**. El procedimiento incluye los siguientes pasos:

1. Edite el archivo de configuración **/etc(exports** en el servidor —servidorA. Compartirá **/usr/local**. Introduzca este texto en el archivo mencionado.

```
#  
/usr/local      clienteA(rw,root_squash)  *(ro)
```

2. Guarde sus cambios en el archivo cuando termine y salga del editor de texto.
3. Enseguida necesitará asegurarse de que el *portmap* está corriendo. Si no lo está, inícielo.

```
[root@servidorA ~]# service portmap status  
portmap is stopped  
[root@servidorA ~]# service portmap start  
Starting portmap: [ OK ]
```

SUGERENCIA En un sistema SuSE, el equivalente de los comandos anteriores sería **rcportmap status** y **rcportmap start**. Y en un sistema que no tiene el comando **service**, busque un archivo posiblemente llamado **portmap** dentro del directorio **/etc/init.d**. Podrá ejecutar manualmente el archivo con las opciones **status** o **start** para controlar el servicio **portmap** con la siguiente instrucción

```
[root@servidorA ~]# /etc/init.d/portmap status
```

4. Luego inicie el servicio **nfs**, el cual activará el resto de los servicios auxiliares que necesita.

```
[root@servidorA ~]# service nfs start  
Starting NFS services: [ OK ]  
Starting NFS quotas: [ OK ]  
Starting NFS daemon: [ OK ]  
Starting NFS mountd: [ OK ]
```

Es habitual que el script de inicio **nfs** le permita saber si completó o falló el inicio. El resultado mostrado arriba nos dice que todo marchó bien.

5. Para verificar que sus exportaciones están configuradas de manera correcta, utilice el comando **showmount**:

```
[root@servidorA ~]# showmount -e localhost
```

6. Si no ve los sistemas de archivos que pone en `/etc/exports`, revise `/var/log/messages` buscando cualquier salida que `nfsd` o `mountd` pudieran haber hecho. Si necesita hacer cambios a `/etc/exports`, ejecute `service nfs reload` o `exportfs -r` cuando termine y, por último, ejecute `showmount -e` para asegurarse de que los cambios surtieron efecto.
7. Ahora que ya tiene configurado el servidor, es momento de poner en marcha al cliente. Primero vea si está funcionando el mecanismo `rpc` entre el cliente y el servidor. De igual forma, utilice el comando `showmount` para verificar que el cliente puede ver los recursos compartidos. Si el cliente no puede, quizás tenga un problema en la red o un problema de permisos en el servidor. Desde clienteA emita el comando

```
[root@clienteA ~]# showmount -e servidorA  
Export list for servidorA:  
/usr/local (everyone)
```

8. Una vez que haya verificado que puede ver los recursos compartidos desde el cliente, es momento de ver si puede montar con éxito un sistema de archivos. Primero genere el punto de montaje `/mnt/usr/local` y luego utilice el comando `mount` como sigue:

```
[root@clienteA ~]# mkdir /mnt/usr/local  
[root@clienteA ~]# mount -o rw,bg,intr,soft servidorA:/usr/local /mnt/usr/local
```

9. Puede utilizar el comando `mount` para ver sólo los sistemas de archivos tipo NFS que están montados en clienteA. Escriba

```
[root@clienteA ~]# mount -t nfs
```

10. Si estos comandos tienen éxito, puede añadir el comando `mount` con sus opciones al archivo `/etc/fstab` de manera que obtenga acceso al sistema remoto desde que reinicia.

```
servidorA:/usr/local      /mnt/usr/local      nfs      rw,bg,intr,soft 0 0
```

USOS COMUNES DE NFS

Las siguientes ideas son, desde luego, sólo ideas. Es casi seguro que usted tendrá sus propias razones para compartir discos a través de NFS.

- ▼ **Para almacenar programas populares.** Si está acostumbrado a Windows, quizás haya trabajado con aplicaciones que se rehusan a ser instaladas en recursos compartidos remotos. Por una u otra razón, estos programas quieren que cada sistema tenga su propia copia del software, una molestia, sobre todo si tiene muchas máquinas que necesitan el software. Linux (y UNIX, en general) rara vez tiene tales condiciones prohibitivas para la instalación de software en discos compartidos en red (las excepciones más comunes son bases de datos de alto desempeño). Así, muchos sitios instalan software de uso frecuente en una partición especial que se exporta a todos los anfitriones en una red.

- **Para almacenar directorios iniciales.** Otro uso común de las particiones de NFS es el almacenamiento de directorios iniciales. Al colocar estos directorios en particiones que pueden montarse con NFS, es posible la configuración de Automounter y NIS o LDAP de manera que los usuarios puedan registrarse desde cualquier máquina en la red al tiempo que tienen a su disposición su propio directorio de inicio. Los sitios heterogéneos utilizan con frecuencia esta configuración de manera que los usuarios puedan moverse sin dificultades de una variante de UNIX a otra sin tener que llevar sus datos consigo.
- ▲ **Para almacenes de correo compartido.** Un directorio que resida en el servidor de correo se puede usar para almacenar todos los buzones de los usuarios y se puede exportar mediante NFS hacia el resto de los anfitriones en la red. En este arreglo los lectores de correo de UNIX tradicionales pueden leer los mensajes en forma directa del almacén contenido en el NFS compartido. En el caso de sitios grandes con tráfico pesado de correo electrónico, varios servidores se pueden usar para proporcionar buzones POP3 y todos los buzones pueden residir en un NFS común compartido que sea accesible para todos los servidores.

RESUMEN

En este capítulo vimos el proceso de poner en marcha un servidor y un cliente de NFS. Ello requiere de muy poca configuración del lado del servidor. El lado del cliente requiere un poquito más de trabajo para su configuración. Pero, en general, el proceso de poner en marcha y ejecutar NFS es relativamente fácil. He aquí algunos puntos para recordar:

- ▼ NFS ha estado entre nosotros por algún tiempo y, como tal, ha pasado por varias revisiones en las especificaciones del protocolo. Las revisiones son en su mayor parte, compatibles con el pasado, y cada revisión posterior puede respaldar clientes utilizando versiones más antiguas.
- NFS versión 4 es la última revisión y contiene muchas mejoras y características que antes no estaban disponibles. Al momento de escribir esta obra la industria presentaba cierta lentitud en la adopción de esta versión, debido quizás a que todos están esperando que alguien la adopte primero y descubra y enmiente las fallas que deban hacerse. No obstante, está ganando popularidad en forma gradual.
- Las versiones anteriores de NFS (las versiones 2 y 3) están implementadas como un protocolo sin estado. Es decir, los clientes no pueden distinguir entre un servidor caído o uno lento; así, la recuperación es automática cuando el servidor regresa en línea (en la situación inversa, cuando el cliente es el que se cae mientras que el servidor se mantiene de pie, la recuperación también es automática).
- ▲ Los procesos clave en NFSv2 y NFSv3 son **rps.statd**, **rpc.quotad**, **rpc.mountd** y **rpc.nfsd**. Casi todas estas funciones han sido llevadas hacia la versión NFSv4.

NFS es una poderosa herramienta para compartir volúmenes de almacenamiento entre los clientes de una red. Le recomendamos que experimente con esta tecnología antes de utilizarla para atender las necesidades de recursos compartidos en su ambiente.

CAPÍTULO 23



Network Information Service (NIS)

El Network Information Service (NIS) (Servicio de información en red) le facilita las cosas si quiere compartir datos críticos almacenados en archivos planos entre varios sistemas en una red. Es usual que los archivos `/etc/passwd` y `/etc/group` sean compartidos con NIS pues, en condiciones ideales, éstos deberían perdurar de manera uniforme en todos los anfitriones. Ofrecer estos archivos mediante NIS permite que cualquier máquina cliente en la red, que esté bien configurada, acceda a los datos contenidos en estos archivos compartidos y utilice las versiones en red como si fueran extensiones de las versiones locales. Sin embargo, NIS no está limitada a compartir sólo estos dos archivos. Cualquier otro archivo tabular puede ser compartido con NIS si cuando menos una columna tiene un valor único en todo el archivo. Ese tipo de archivos son muy comunes en los sistemas Linux/UNIX; por ejemplo, el archivo de alias de Sendmail, los archivos Automounter o el archivo `/etc/services`.

El beneficio principal logrado por el uso de NIS es el hecho de que usted puede mantener una copia central de los datos, y en el momento en que éstos se actualicen, automáticamente dicha actualización se propaga a todos los usuarios de la red. Las características de NIS ayudan a sus usuarios a presentar la apariencia de un sistema más uniforme, sin importar en qué anfitrión estén trabajando.

Si usted llega después de recorrer ambientes de Windows, podría pensar que NIS es la solución que proporciona Linux/UNIX de algunos de los servicios ofrecidos por Active Directory. Por supuesto que NIS es una tecnología mucho más antigua y como tal no intenta resolver (o crear ☺) la abundancia de problemas que obstruyen Active Directory.

En este capítulo exploraremos NIS, cómo trabaja y cómo lo puede ayudar. Luego explicaremos cómo actualizar las partes del cliente y el servidor de la configuración de NIS. Al final analizaremos algunas de las herramientas relacionadas con NIS.

CONTENIDO DE NIS

El Servicio de información en red en realidad sólo es una base de datos sencilla que los clientes pueden consultar. Contiene un conjunto de tablas independientes. Cada tabla se crea a partir de archivos de texto plano, como el archivo `/etc/passwd`, que es de naturaleza tabular y tiene cuando menos una columna que es única para cada renglón (lo que en una base de datos es una pareja llave/valor). NIS lleva control de estas tablas por su nombre y permite que las consultas ocurran de una de dos maneras:

- ▼ Listando la tabla en su totalidad.
- ▲ Extrayendo un registro específico que coincide con una llave buscada.

Una vez que se establecen las bases de datos en el servidor, los clientes pueden consultarlas buscando registros específicos en éstas. Es común que esto suceda cuando un cliente se configura para que se asome al *mapa* NIS cuando un registro no puede encontrarse en la base de datos local del cliente. Un anfitrión puede tener un archivo que sólo contiene los registros necesarios para trabajar en modo monousuario (cuando no hay conectividad en red), por ejemplo, el archivo `/etc/passwd`. Cuando un programa hace una solicitud para buscar información sobre contraseñas de usuario, el cliente revisa su archivo local `passwd` y verifica que el usuario no esté ahí; entonces el cliente hace una solicitud al servidor NIS para que busque el registro correspondiente en la tabla `passwd`. Si el NIS tiene un registro coincidente, éste se devuelve al cliente y luego al programa

que en primer lugar solicitó la información. El programa no sabe que se usó NIS. Lo mismo sucede si el mapa NIS devuelve una respuesta informando que la contraseña del usuario no existe. La información le sería devuelta al programa sin que éste supiera cuántas actividades se llevaron a cabo para ello.

Desde luego, esto sólo se aplica para todos aquellos archivos que compartimos mediante NIS. Otros populares archivos que se comparten son `/etc/group` y `/etc/hosts`.

NOTA Aunque técnicamente es correcto referirse a las tablas de NIS como una base de datos, es más usual que se les llame mapas (en este contexto, estamos “mapeando” llaves a valores). Si utilizamos el archivo `/etc/passwd` como ejemplo, lo que estamos haciendo es relacionar el nombre de acceso del usuario (el cual sabemos que es siempre único) con el resto del registro, que es la contraseña.

Enseguida mostramos una lista con algunos demonios y procesos asociados a NIS:

- ▼ **ypserv** Este demonio se ejecuta en el servidor NIS. Escucha y atiende las consultas de los clientes ofreciendo las respuestas pertinentes.
- **ypxfrd** Este demonio se usa para propagar y transferir las bases de datos NIS entre servidores esclavos.
- ▲ **ypbind** Este es el componente NIS del lado del cliente. Es responsable de encontrar un servidor NIS al que se le pueda consultar información. Este demonio liga clientes de NIS con un dominio NIS. Debe estar funcionando en máquinas que ejecuten los programas NIS clientes.

Los servidores NIS

El sistema NIS puede tener solamente un servidor que tiene la autoridad donde los archivos de datos originales se almacenan (esto es algo similar a DNS). A este servidor NIS que tiene la autoridad se le llama servidor NIS *maestro*. Si su organización es bastante grande, quizás necesite distribuir la carga entre más de una máquina. Esto puede lograrse poniendo en marcha uno o más servidores NIS *secundarios* (*esclavos*). Además de ayudarle a distribuir la carga, los servidores secundarios también proporcionan un mecanismo para enfrentar de mejor manera las fallas. El servidor NIS secundario puede continuar respondiendo consultas aun cuando el servidor maestro u otros servidores secundarios estén caídos.

NOTA Un servidor puede ser, al mismo tiempo, cliente y servidor.

Los servidores NIS secundarios reciben actualizaciones cuando el servidor NIS primario se actualiza, de manera que los maestros y los esclavos permanezcan en sincronía. El proceso de mantener a los servidores secundarios en sincronía con el servidor primario se conoce como *propagación hacia servidores* (*server push*). Como parte de su rutina de actualización, el maestro NIS también proporciona una copia de los archivos mapa al servidor secundario. Al recibir estos archivos, el servidor secundario a su vez actualiza sus bases de datos. El servidor NIS maestro no se considera a sí mismo completamente actualizado hasta que los servidores secundarios también están actualizados.

NOTA En NIS también existe un mecanismo de *extracción desde servidores* (*server pull*). Sin embargo, con frecuencia esta solución se reserva para configuraciones más complejas, como en el caso de tener cientos de servidores esclavos. En una red de menores dimensiones, esto no debería ser un problema.

Dominios

Los servidores NIS primarios establecen *dominios* que son similares a los dominios de un controlador de dominios (DC, *domain controller*) en Windows. Una importante diferencia es que el dominio NIS no requiere que el administrador del servidor NIS permita en forma explícita que el cliente se una al dominio (tenga presente que el modelo NIS supone que todos los clientes son miembros del mismo dominio administrativo y, por tanto, su gestión la realizan los mismos administradores del sistema). Además, el servidor NIS sólo centraliza información / datos; no realiza por sí solo la autenticación; esta tarea la delega en otras rutinas del sistema. El proceso de autenticar usuarios se deja a cada anfitrión individual; NIS tan sólo proporciona una lista centralizada de usuarios.

SUGERENCIA Debido a que los dominios NIS deben “bautizarse” con nombres, es una buena práctica, aunque no obligatoria, utilizar nombres diferentes de aquellos que se usan para los nombres de dominio DNS. Se le facilitará la vida cuando dialogue sobre sus dominios de red con sus colegas administradores cuando todos sepan diferenciar entre unos y otros.

CONFIGURACIÓN DEL SERVIDOR NIS MAESTRO

Es usual que las distribuciones de Linux ya tengan instalado el software NIS que va del lado del cliente como parte de la instalación inicial del sistema operativo. Este arreglo facilita la puesta en marcha de cualquier sistema como un cliente de NIS desde el arranque, algunas distribuciones incluso le darán oportunidad de configurar una máquina para que utilice NIS al momento de instalar el sistema operativo.

Quizá necesite instalar manualmente el componente NIS que va en el lado del servidor debido a que no todo sistema necesita actuar como servidor NIS. Casi siempre se trata de un proceso sin sobresaltos. El software requerido puede descargarse con facilidad del almacén de software que corresponde a su distribución (es decir, del sitio Web o del medio de instalación).

Después de instalar el software del servidor NIS, lo que queda por hacer es habilitar el servicio (si es que no está habilitado todavía) y configurarlo. Para asegurarse de que el servidor NIS (*ypserv*) inicia en forma automática cuando arranca el sistema, puede utilizar la herramienta **chkconfig**.

Primero instalaremos el software NIS que va del lado del servidor. En un sistema Fedora Core y en la mayoría de los sistemas Linux basados en RPM, el paquete de software que proporciona la funcionalidad del servidor NIS se llama *ypserv*.rpm* (donde * representa el número de la versión disponible).

En este ejemplo utilizaremos el programa Yum para descargar e instalar con celeridad el paquete desde Internet. Introduzca el siguiente comando Yum:

```
[root@servidorA ~]# yum install ypserv
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
Installing: ypserv                                     #####
Installed: ypserv.i386 0:*
Complete!
```

SUGERENCIA En un sistema Fedora puede realizar la instalación directamente desde el medio de instalación; puede encontrar el paquete en el directorio /media_mount_point/Fedora/RPMS. En un sistema SuSE, puede llevar a cabo dicha instalación por medio de la utilidad Yast si ejecuta **yast -i ypserv**.

Una vez que NIS está instalado y listo, necesitará configurarlo. Hay cuatro pasos para lograrlo:

1. Establezca el nombre del dominio.
2. Inicie el demonio **ypserv** para empezar NIS.
3. Edite **Makefile**.
4. Ejecute **ypinit** para crear las bases de datos.

Estos pasos se examinan a detalle en la siguiente sección.

Establecimiento del nombre del dominio

Utilice el comando **domainname** para establecer el nombre del dominio. Suponga que vamos a poner en marcha un dominio NIS llamado nis.ejemplo.org.

Primero utilice el comando **domainname** para ver el dominio NIS que ya está definido en el sistema. Escriba

```
[root@servidorA ~]# domainname  
(none)
```

Verá que en este momento no hay ninguno. Ahora sigamos adelante y definamos el dominio NIS como sigue:

```
[root@servidorA ~]# domainname nis.ejemplo.org
```

Ejecute el comando **domainname** una vez más para ver sus cambios. Escriba

```
[root@servidorA ~]# domainname  
nis.ejemplo.org
```

Para que el nombre de dominio NIS perdure cuando tenga que reiniciar el equipo, en un sistema Fedora Core y en la mayoría de los sistemas tipo Red Hat defina una variable llamada **NISDOMAIN** en el archivo **/etc/sysconfig/network**. Abra el archivo para editar y añada una línea como la siguiente al final del archivo:

```
NISDOMAIN=nis.ejemplo.org
```

Usaremos el comando **echo** para hacer los cambios mencionados en el archivo **/etc/sysconfig/network**. Escriba

```
[root@servidorA ~]# echo "NISDOMAIN=nis.ejemplo.org" >> /etc/sysconfig/network
```

SUGERENCIA En otras distribuciones de Linux puede obtener el mismo efecto si agrega el comando **domainname** con el valor apropiado para uno de los scripts **rc** que se ejecutan cuando el sistema arranca; por ejemplo, puede editar el **/etc/init.d/ypserv**. Ábralo y busque la línea que contiene **domainname**; si no existe, añada una después de la primera línea. La línea que debe leer es como la siguiente:

```
domainname nis.ejemplo.org
```

Recuerde reemplazar **nis.example.org** con su propio nombre de dominio NIS. El nombre del dominio debe ser configurado antes de que el servidor NIS (**ypserv**) inicie.

El arranque de NIS

El demonio **ypserv** es responsable de manejar las solicitudes de NIS. Es muy fácil iniciar el demonio **ypserv** en un sistema Fedora Core o en uno RHEL. Utilizaremos el comando **service** para iniciarlo aquí. Para otras distribuciones de Linux, puede ejecutar el script de arranque **ypserv** (ubicado en **/etc/init.d/ypserv**) en forma directa, si así lo desea. Para un sistema SuSE utilice el comando **rcypserv** con el parámetro apropiado.

NIS es un servicio basado en RPC, así que debe asegurarse de que el portmap esté activo y corriendo antes de iniciar NIS. Para iniciar el servicio **portmap** en un sistema Fedora, escriba

```
[root@servidorA ~]# service portmap start
```

En nuestro sistema de muestra iniciaremos el sistema NIS como sigue:

```
[root@servidorA ~]# service ypserv start
```

Para confirmar que el servicio **ypserv** se ha registrado en forma correcta con el portmap utilice el comando **rpcinfo** como sigue:

```
[root@servidorA ~]# rpcinfo -p
program vers proto port
 100000    2   tcp    111  portmapper
 100000    2   udp    111  portmapper
 100024    1   udp   32768  status
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
 100004    2   tcp     884  ypserv
 100004    1   tcp     884  ypserv
```

Si necesita detener el servidor NIS en cualquier momento, puede hacerlo con el comando

```
[root@servidorA ~]# service ypserv stop
```

Edición del archivo Makefile

Hasta ahora ha visto el uso del comando **make** para compilar programas de capítulos anteriores. Sin embargo, dicho comando no es el que hace la compilación, solamente lleva registro de los

archivos que necesitan compilarse y luego invoca los programas necesarios para realizar la compilación. El archivo que en realidad contiene las instrucciones para **make** se llama *makefile*.

La herramienta **make** es eficiente porque los programas que invoca pueden ser definidos según sea necesario. Por ejemplo, puede asignar su compilador preferido en lugar del que viene definido en una particular distribución de Linux. Cuando **make** se da cuenta de que la fecha y hora de un archivo se modifican, considera que su contenido ha cambiado. Si el archivo en efecto se ha modificado, **make** supone que el archivo necesita ser recompilado.

Poner este concepto a trabajar en NIS es bastante claro y directo. En este caso hay un conjunto de archivos de texto que necesitan ser convertidos a un formato de base de datos. Necesitamos una herramienta que se encargue de reconvertisir cualesquiera archivos que hayan sido modificados, ¡y **make** hace este trabajo a la perfección!

Si entra al directorio **/var/yp** verá un archivo llamado **Makefile** (así es, una sola palabra). Este archivo lista los archivos que se comparten vía NIS, además de otros parámetros adicionales sobre cómo se comparten y qué tanto cada uno de ellos se comparte. Abra el archivo **Makefile** en su editor de textos favorito y podrá ver todas las opciones que puede configurar. Veamos las opciones que se aplican a Linux en este archivo.

Pero antes de proceder, debería hacer una copia de seguridad del archivo **Makefile** original. Utilice el comando **copy** (**cp**) para lograrlo:

```
[root@servidorA ~]# cp /var/yp/Makefile /var/yp/Makefile.original
```

En la siguiente sección describiremos algunas directrices del archivo **Makefile** que son de interés particular para nosotros en este capítulo. Algunas secciones de este archivo se refieren aquí, incluyendo sus comentarios para mayor claridad.

Designación de servidores esclavos: NOPUSH

Si planea contar con servidores NIS esclavos, necesitará decirle al servidor NIS maestro que propague los mapas resultantes entre los servidores esclavos. Cambie la variable **NOPUSH** a falso si quiere servidores esclavos.

NOTA Si no requiere servidores esclavos ahora pero piensa que podría utilizarlos después, cambie esta opción cuando añada dichos servidores.

```
# Si sólo tenemos un servidor, no necesitamos propagar los mapas entre los
# servidores esclavos (NOPUSH=true). Si tiene servidores esclavos, cambie este parámetro
# a "NOPUSH=false" y ponga todos los nombres de anfitrión de sus servidores esclavo en
el archivo
# /var/yp/ypservers.
NOPUSH=true
```

Recuerde listar los nombres de anfitrión de sus servidores esclavo en el archivo **/var/yp/ypservers**. Y para cada nombre de anfitrión que liste ahí, asegúrese de listar la entrada correspondiente en el archivo **/etc/hosts**.

UID y GID mínimas: MINUID y MINGID

Cuando añade cuentas, la mínima UID y GID creadas en los archivos **/etc/passwd** y **/etc/group** serán diferentes dependiendo de su distribución de Linux. Asegúrese de establecer los valores

mínimos de UID y GID que está dispuesto a compartir mediante NIS. Es obvio que usted no desea compartir la entrada raíz mediante NIS, así que el mínimo nunca debería ser cero.

```
# No incluimos entradas de contraseñas con UID bajas (las entradas root y system)
# en la base de datos de contraseñas de NIS, por seguridad. MINUID es la uid más
# baja que se incluirá en los mapas de contraseñas. Si usted crea mapas ocultos
# (shadow maps), la ID de usuario (UserID) para una entrada oculta se toma del
# archivo passwd. Si no se encuentra una entrada, esta entrada oculta se
# ignora.
# MINGID es la gid más baja que se incluirá en los mapas de grupo.
MINUID=500
MINGID=500
```

Combinación de contraseñas ocultas con contraseñas reales: **MERGE_PASSWD**

Para que NIS se pueda usar en la autenticación de usuarios necesitará permitir que las entradas de contraseñas con encryptación sean compartidas a través de NIS. Si está utilizando contraseñas ocultas (shadow passwords), NIS le ayudará a manejar esta situación en forma automática tomando el campo encryptado del archivo **/etc/shadow** para luego combinarlo con la copia compartida NIS del archivo **/etc/passwd**. A menos que haya una razón específica por la que no quiera habilitar que las contraseñas con encryptación sean compartidas, deje MERGE_PASSWD intacto.

```
# Debemos combinar el archivo passwd con el archivo shadow ?
# MERGE_PASSWD=true|false
MERGE_PASSWD=true
```

Combinación de contraseñas ocultas de grupo con grupos reales: **MERGE_GROUP**

El archivo **/etc/group** permite la aplicación de contraseñas a las configuraciones de grupo. Como el archivo **/etc/group** necesita estar disponible públicamente para su lectura, algunos sistemas ofrecen soporte para archivos de grupos ocultos, que son de naturaleza similar a los archivos de contraseñas ocultas. A menos que tenga un archivo de grupos oculto, necesita establecer el valor de MERGE_GROUP a falso.

```
# Debemos combinar el archivo group con el archivo gshadow ?
# MERGE_GROUP=true|false
MERGE_GROUP=false
```

Designación de nombres de archivo

La siguiente sección del archivo **Makefile** muestra los archivos que están configurados en forma predeterminada para ser compartidos mediante NIS. Sin embargo, sólo porque están listados aquí no significa que sean compartidos en forma automática. Este listado establece variables para uso posterior en el archivo **Makefile**.

YPPWDDIR = /etc

Esta variable (**YPPWDDIR**) especifica la ubicación de los archivos passwd, group y shadow.

YPSRCDIR = /etc

La variable **YPSRCDIR** por lo general se utiliza para especificar la ubicación del directorio de otros archivos fuente para NIS. Casi siempre se utiliza para archivos relacionados con la red tales como los archivos de anfitriones, protocolos y servicios. La variable se ocupa ampliamente en el resto del archivo para especificar la ubicación de otros archivos que podrían ser de interés.

El listado que sigue muestra el uso real de las variables **YPPWDIR** y **YPSRCDIR** en el archivo **Makefile**.

```
# Estos son los archivos a partir de los cuales se construyen las bases de datos
# NIS. Puede editarlos a su gusto en caso de que desee mantener los archivos NIS
# fuente separados de los archivos de configuración real de su servidor NIS.
#
GROUP      = $(YPPWDDIR)/group
PASSWD     = $(YPPWDDIR)/passwd
SHADOW     = $(YPPWDDIR)/shadow
GSHADOW    = $(YPPWDDIR)/gshadow
ADJUNCT    = $(YPPWDDIR)/passwd.adjunct
#ALIASES   = $(YPSRCDIR)/aliases # aliases pueden estar en /etc or /etc/mail
ALIASES    = /etc/aliases
ETHERS     = $(YPSRCDIR)/ethers      # ethernet addresses (para rarpd)
BOOTPARAMS = $(YPSRCDIR)/bootparams # para iniciar equipos Sun (bootparamd)
HOSTS      = $(YPSRCDIR)/hosts
NETWORKS   = $(YPSRCDIR)/networks
PRINTCAP   = $(YPSRCDIR)/printcap
PROTOCOLS  = $(YPSRCDIR)/protocols
PUBLICKEYS = $(YPSRCDIR)/publickey
RPC        = $(YPSRCDIR)/rpc
SERVICES   = $(YPSRCDIR)/services
NETGROUP   = $(YPSRCDIR)/netgroup
NETID      = $(YPSRCDIR)/netid
AMD_HOME   = $(YPSRCDIR)/amd.home
AUTO_MASTER = $(YPSRCDIR)/auto.master
AUTO_HOME  = $(YPSRCDIR)/auto.home
AUTO_LOCAL = $(YPSRCDIR)/auto.local
TIMEZONE   = $(YPSRCDIR)/timezone
LOCALE     = $(YPSRCDIR)/locale
NETMASKS   = $(YPSRCDIR)/netmasks
```

Lo que se comparte: la instrucción all

En la siguiente instrucción del archivo **Makefile**, todos aquellos mapas listados después de **all**: son los mapas que se compartirán:

```
all: passwd group hosts rpc services netid protocols mail \
      # netgrp shadow publickey networks ethers bootparams printcap \
      # amd.home auto.master auto.home auto.local passwd.adjunct \
      # timezone locale netmasks
```

Note que en el fragmento anterior se utilizan diagonales invertidas (\) para marcar la continuación de la línea. Esto hace que el programa **make** considere a la instrucción como una sola línea, aunque en realidad esté conformado por cuatro. Además, note que el segundo, tercer y cuarto renglones empiezan con un signo de número (#), lo que significa que el resto de la línea es inerte porque es un comentario.

Según este formato, podrá ver que los mapas configurados son **passwd**, **group**, **hosts**, **rpc**, **services**, **netid**, **protocols** y **mail**. Estas entradas corresponden a nombres de archivo listados en la sección anterior del archivo **Makefile**. Desde luego, no en todos los sitios se querrá compartir estas entradas, o quizás quieran mapas adicionales (tales como los archivos Automounter: **auto.master** y **auto.home**). Para cambiar cualesquiera de los mapas que quiera compartir, modifique la línea de tal manera que los mapas que *no quiera* compartir estén listados después del símbolo #.

Por ejemplo, digamos que usted sólo quiere que los mapas **passwd** y **group** sean compartidos mediante NIS. De este modo, tendría que cambiar la instrucción **all:** como sigue:

```
all: passwd group \
    # hosts rpc services protocols netgrp mail \
    # shadow publickey networks ethers bootparams amd.home \
    # passwd.adjunct
```

Note que en la instrucción **all:** el orden de los mapas no importa. La posición de las entradas posteriores tan sólo hace que éstas sean fáciles de leer.

El uso de **ypinit**

Una vez que tiene listo el archivo **Makefile**, necesita inicializar el servidor YP (NIS) utilizando el comando **ypinit**.

NOTA Recuerde que el dominio NIS ya necesita estar definido antes de que funcione el comando **ypinit**. Esto se logra con la utilidad **domainname**, como se mostró en este capítulo en la sección “Establecimiento del nombre del dominio”.

```
[root@servidorA ~]# /usr/lib/yp/ypinit -m
```

Aquí la opción **-m** especifica que **ypinit** debe poner en marcha el servidor NIS como un servidor maestro. Suponiendo que ejecutamos este comando en nuestro sistema de muestra llamado **servidorA**, veríamos que el sistema responde como sigue:

```
At this point, we have to construct a list of the hosts which will run NIS
servers. serverA.example.org is in the list of NIS server hosts. Please con-
tinue to add the names for the other hosts, one per line. When you are done
with the list, type a <control D>.
```

```
next host to add: servidorA.ejemplo.org
next host to add:
```

En este punto debe continuar proporcionando los nombres de los servidores NIS secundarios si es que va a tenerlos. Presione CTRL-D cuando termine de añadir todos los servidores necesarios.

Estas entradas serán colocadas en el archivo `/var/yp/ypservers`; en caso de ser necesario, puede cambiarlas editando más tarde ese archivo.

Enseguida se le pedirá que confirme si la información proporcionada es correcta. La lista actual de servidores NIS se ve como esto:

```
servidorA.ejemplo.org
Is this correct? [y/n: y] y
We need a few minutes to build the databases...
Building /var/yp/nis.ejemplp.org/ypservers...
gethostbyname(): Success
Running /var/yp/Makefile...
gmake[1]: Entering directory `/var/yp/nis.ejemplo.org'
Updating passwd.byname...
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating passwd.
byuid...
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating group.
byname...
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
servidorA.ejemplo.org has been set up as a NIS master server.
Now you can run ypinit -s servidorA.ejemplo.org on all slave servers.
```

(Por ahora ignore los mensajes de error que pudieran haber resultado de este comando. Los posibles errores se tratan con mayor detalle en la siguiente sección.)

Una vez terminado, `ypinit` ejecutará el programa `make` en forma automática a fin de construir los mapas y propagarlos entre los servidores secundarios que hubiera especificado.

Este pudiera ser un buen momento para asegurarse de que `portmap` y los servicios del servidor NIS están funcionando. Inícielos ahora con los siguientes comandos en caso de que no lo estén:

```
[root@servidorA ~]# service portmap start
Starting portmap:                                     [ OK ]
[root@servidorA ~]# service ypserv start
Starting YP server services:                         [ OK ]
```

Errores del archivo Makefile

Examine los errores que pudieron haber surgido al ejecutar el comando `ypinit` en la sección anterior. Casi seguramente se trató de errores no fatales.

Si cometió un error en el archivo **Makefile**, quizás obtenga un error cuando `ypinit` ejecuta el programa `make`. Si ve un error como el siguiente,

```
gmake[1]: *** No rule to make target '/etc/shadow', needed by 'passwd.byname'.
Stop.
```

no se preocupe. Significa que quiere compartir un archivo que no existe (en esta muestra de mensaje de error se trata del archivo `/etc/shadow`). La solución está en crear el archivo o editar **Makefile** para eliminar o corregir la referencia al archivo inexistente. (Vea la sección previa "Lo que se comparte: la instrucción `all`".)

Otro mensaje de error común es

```
failed to send 'clear' to local ypserv: RPC: Program not registered  
Updating passwd.byuid...  
failed to send 'clear' to local ypserv: RPC: Program not registered  
gmake[1]: *** No rule to make target '/etc/gshadow', needed by 'groupbyname'.  
Stop.  
gmake[1]: Leaving directory '/var/yp/serverA.example.org'
```

De hecho, en esta muestra hay dos mensajes de error. Puede ignorar el primero, que indica que el servidor NIS aún no se ha iniciado. El segundo mensaje de error es sobre el mismo asunto descrito en la muestra anterior. Una vez que lo haya corregido, escriba el siguiente comando para reconstruir los mapas, como se describe en la siguiente sección:

```
[root@servidorA ~]# cd /var/yp ; make
```

Actualización de mapas NIS

Si tuvo que actualizar los archivos que configuró para que sean compartidos por NIS con el resto de su red, necesitará reconstruir los archivos de mapas (por ejemplo, quizás añadió un usuario al archivo `/etc/passwd` central). Para reconstruir los mapas, utilice el siguiente comando `make`:

```
[root@servidorA ~]# cd /var/yp ; make
```

CONFIGURACIÓN DE UN CLIENTE DE NIS

Afortunadamente, ¡los clientes de NIS son mucho más fáciles de configurar que los servidores NIS! Todo lo que debe hacer para poner en funcionamiento un cliente de NIS es llevar a cabo lo siguiente:

1. Edite el archivo `/etc/yp.conf`.
2. Configure el script de arranque.
3. Edite el archivo `/etc/nsswitch.conf`.

Edición del archivo `/etc/yp.conf`

El archivo `/etc/yp.conf` contiene la información necesaria para el demonio que va del lado del cliente, `ypbind`, a fin de que éste inicie y encuentre el servidor NIS. Necesita tomar una decisión respecto de la forma como el cliente buscará al servidor, ya sea utilizando una difusión de señal o especificando el nombre de anfitrión del servidor.

La técnica de difusión es apropiada cuando necesita mover un cliente entre varias subredes y usted no quiere verse obligado a reconfigurar al cliente en tanto el servidor NIS exista en la misma subred. La desventaja de esta técnica, claro está, es que necesita asegurarse de que existe un servidor NIS en cada subred.

NOTA Cuando utiliza el método de difusión, debe contar con un servidor NIS en cada subred debido a que, bajo condiciones normales, los routers no redirigirán el tráfico; es decir, la difusión de señales no abarca múltiples subredes. Si no sabe con seguridad que un servidor NIS en particular está en la misma subred, puede saberlo si utiliza la herramienta “ping” dirigida hacia la dirección de difusión (algunos sistemas han habilitado la protección contra ataques pitufo por lo que quizás no respondan al uso de ping, quizás necesite inhabilitar temporalmente la protección para hacer las pruebas como se debe). Si el servidor NIS es uno de los anfitriones que responde, entonces sabe con seguridad que el método de difusión funcionará.

La otra técnica para lograr el contacto cliente-servidor es especificar el nombre del anfitrión del servidor. Este método trabaja bien cuando necesita establecer subredes en su red, pero no necesita un servidor NIS en cada subred. Esto permite que un cliente se mueva hacia cualquier lugar dentro de la red y aun sea capaz de encontrar el servidor NIS; sin embargo, si necesita cambiar a un cliente para que apunte a otro servidor NIS (para balancear la carga de la red, por ejemplo), necesitará realizar esa modificación usted mismo.

- ▼ **Método de difusión** Si elige la técnica de difusión, edite el archivo `/etc/yp.conf` en el cliente de manera que lea como sigue:

```
domain nis.ejemplo.org broadcast
```

donde `nis.ejemplo.org` es el nombre de nuestro dominio NIS de muestra. Recuerde que si necesita soporte a prueba de fallas, necesitará dos servidores NIS en cada subred de manera que la señal de difusión encuentre el segundo servidor.

- ▲ **Método del nombre de anfitrión del servidor** Si quiere especificar el nombre del servidor NIS en forma directa, edite el archivo `/etc/yp.conf` de manera que lea como sigue:

```
domain nis.ejemplo.org server servidora
```

donde `nis.ejemplo.org` es el nombre de nuestro dominio NIS de muestra y `servidora` es el nombre del servidor NIS al que este cliente deberá apuntar.

NOTA Recuerde que también necesita contar con una entrada para `servidora` en el archivo `/etc/hosts`. Cuando se inicia el servidor NIS, quizás no cuente con acceso a DNS todavía y ¡seguro tampoco tendrá acceso a la tabla de anfitriones NIS! Por esta razón, el cliente debe ser capaz de representar el nombre del anfitrión como una dirección IP sin ayuda de ningún otro servicio.

Para habilitar e iniciar `ypbind`

El cliente de NIS ejecuta un demonio llamado `ypbind` a fin de que pueda comunicarse con el servidor. Con frecuencia ello se hace en el script de inicio `/etc/init.d/ypbind`. Revise sus scripts de inicio con el programa `chkconfig` y verifique si `ypbind` iniciará en forma automática en los niveles de ejecución deseados.

- ▼ Para iniciar el demonio sin tener que reiniciar la computadora, utilice el comando:

```
[root@servidorA ~]# service ypbind start
```
- Si necesita detener el funcionamiento del demonio, escriba

```
[root@servidorA ~]# service ypbind stop
```
- ▲ Utilice la utilidad **chkconfig** para habilitar el inicio automático de **ypbind** en los niveles de ejecución 3 y 5. Escriba

```
[root@servidorA ~]# chkconfig --level 35 ypbind on
```

Si no tiene disponible la utilidad **service** en su sistema, puede ejecutar a mano los scripts de inicio añadiendo el parámetro apropiado. Por ejemplo, para iniciar el servicio **ypbind** escribiría

```
[root@servidorA ~]# /etc/init.d/ypbind start
```

Edición del archivo /etc/nsswitch.conf

El archivo **/etc/nsswitch.conf** es responsable de decirle al sistema el orden en el cual deberá buscar la información. El formato del archivo es como sigue:

```
nombrearchivo: nombreservicio
```

donde **nombrearchivo** es el nombre del archivo que necesita referenciar, y **nombreservicio** es el nombre del servicio que habrá de utilizarse para encontrar el archivo. Es posible listar varios servicios, separados por espacios. He aquí ejemplos de algunos servicios válidos:

files	Utiliza el archivo real en este mismo anfitrión.
yp	Utiliza NIS para realizar la búsqueda.
nis	Utiliza NIS para realizar la búsqueda (nis es un alias para yp).
dns	Utiliza DNS para la búsqueda (sólo se aplica en anfitriones).
[NOTFOUND=return]	Detiene la búsqueda.
nis+	Utiliza NIS+ (evite utilizar esta opción debido a que, al momento de escribir esta obra, la implementación NIS+ en Linux todavía tenía un estatus experimental).
ldap	Utiliza el Protocolo ligero de acceso a directorios (LDAP, Lightweight Directory Access Protocol).

Enseguida mostramos un ejemplo de una entrada en el archivo **/etc/nsswitch.conf**:

```
passwd: files nis
```

Esta entrada muestra que las solicitudes de búsqueda para contraseñas (**passwd**) se harán primero en el archivo **/etc/passwd**. Si la entrada solicitada no está ahí, entonces se buscará utilizando NIS.

El archivo `/etc/passwd` ya debe existir y debe contener gran parte de la información necesaria. Quizá tenga que hacer ajustes en el orden en el que los nombres de los servicios se listan en el archivo.

Herramientas GUI para NIS

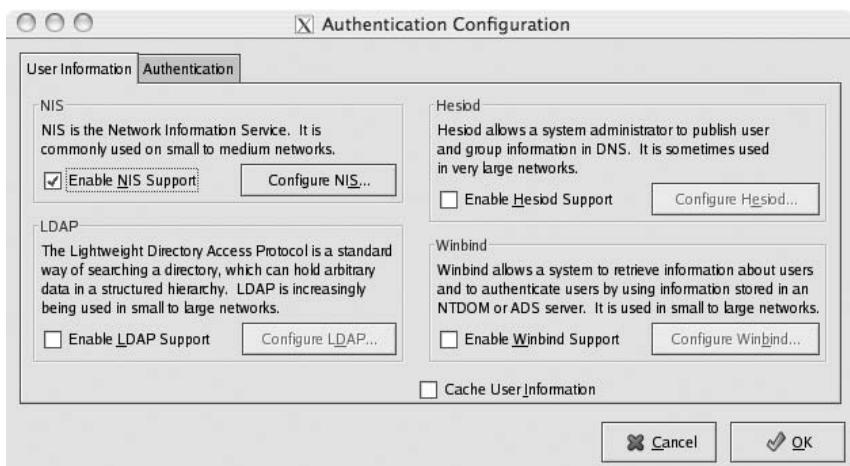
Fedora Core y RHEL tienen algunas herramientas para la GUI (Graphic User Interface, Interfaz gráfica del usuario) que pueden facilitar la configuración de un cliente de NIS. La primera herramienta se basa en ncurses para la línea de comandos y se llama **authconfig**. Se muestra enseguida:



Para iniciarla, escriba

```
[root@servidorA ~] # authconfig
```

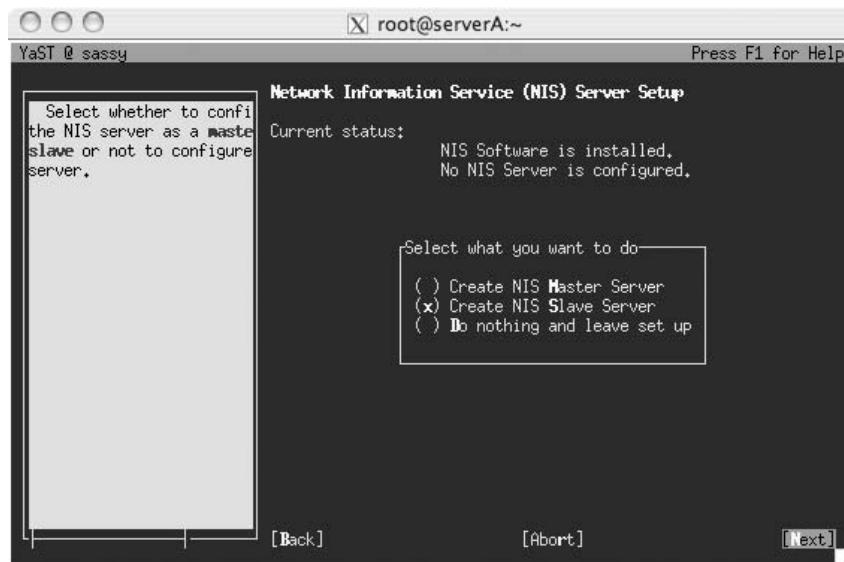
La segunda herramienta es **system-config-authentication**, mostrada enseguida:



Esta herramienta requiere que tenga en funcionamiento el ambiente X Window System. Para iniciar la herramienta, escriba

```
[root@servidorA ~]# system-config-authentication
```

SuSE Linux tiene atractivas herramientas para la GUI que facilitan la configuración de un servidor o un cliente de NIS. Enseguida se muestra la herramienta de configuración de un servidor:



Para iniciar esta herramienta, escriba

```
servidorA:~ # yast nis_server  
o  
servidorA:~ # yast2 nis_server
```

NIS trabajando

La ilustración en la figura 23-1 muestra un ejemplo de uso de NIS. En la ilustración se representa el intento de registro de un usuario, antes de que NIS se ponga en marcha. La segunda parte de la ilustración muestra el intento de registro del mismo usuario, una vez que NIS se inicia.

El usuario Fulanito intentará registrarse en su sistema local como un usuario que *no* existe en el archivo de anfitriones local del servidorB /etc/passwd. El intento fracasará.

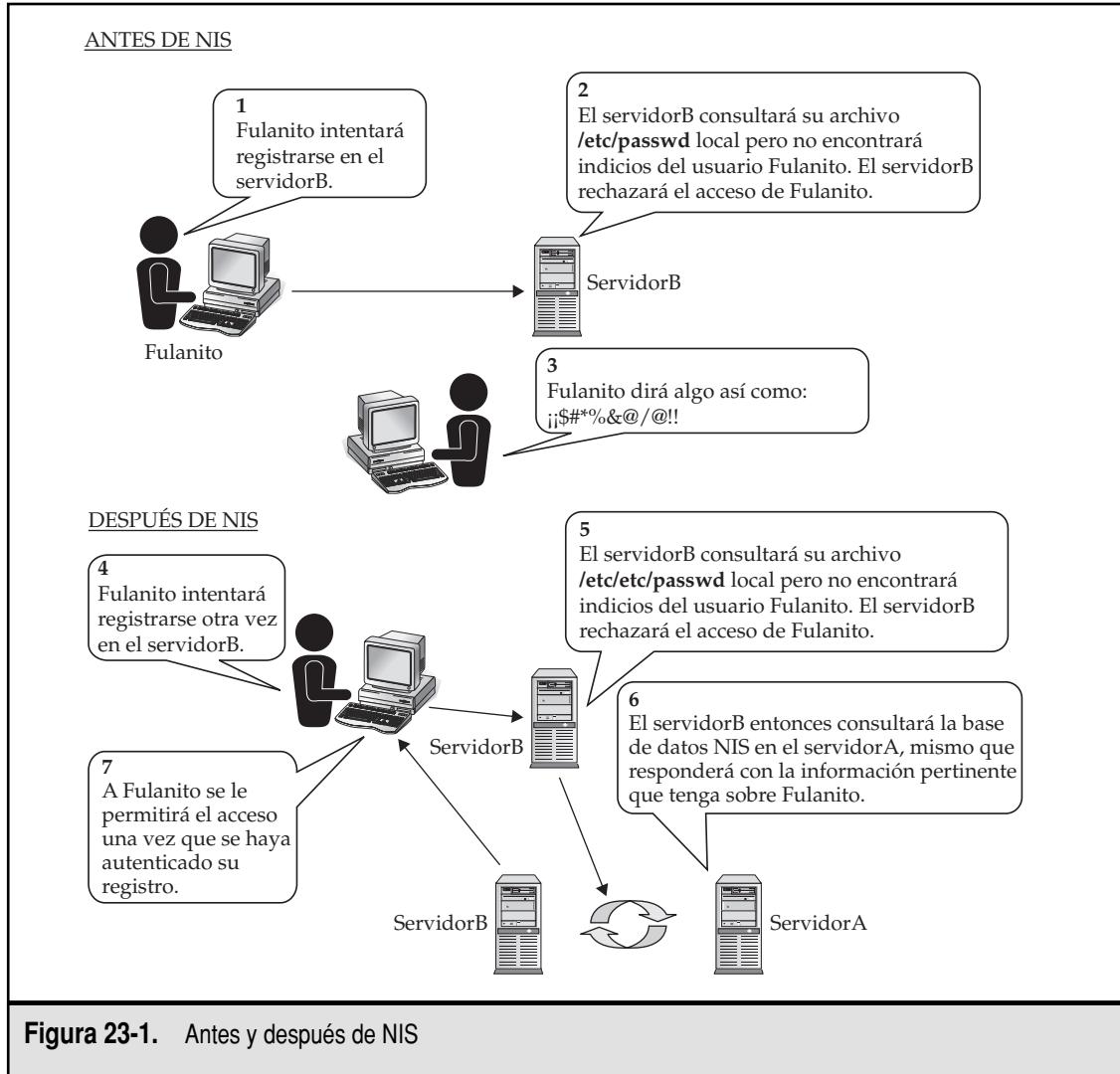


Figura 23-1. Antes y después de NIS

Después de configurar NIS, un intento de registro similar por el mismo usuario Fulanito ahora sí tendrá éxito. Esto se debe a que el servidorB ya se ha configurado como cliente de NIS del servidorA. El cliente, servidorB, de nuevo verificará su archivo /etc/passwd local buscando indicios del usuario Fulanito pero, al no encontrar nada, inmediatamente consultará al servidor NIS maestro en el servidorA. Como el servidor NIS maestro sí tiene conocimiento sobre dicho usuario, entregará esa información al servidorB y este último realizará la autenticación real.

Pruebas de la configuración del cliente de NIS

Después de la correcta configuración de los archivos `/etc/yp.conf` y `/etc/nsswitch.conf`, y después de que el demonio `ypcat` se puso en marcha, debería ser posible utilizar el comando `ypcat` para vaciar un mapa desde el servidor NIS hasta su pantalla. Para lograrlo, escriba el siguiente comando:

```
[root@servidorA yp]# ypcat passwd
yyang:$1$Yn7AHj/:500:500:Ying Yang:/home/yyang:/bin/bash
rmota:$1$jKv9B:501:501:Rolando Mota:/home/rmota:/bin/bash
```

mismo que vaciará el mapa `passwd` en su pantalla, *en caso* de que lo esté compartiendo vía NIS, claro está. Si no lo está compartiendo, escoja un mapa qué sí este compartido y utilice el comando `ypcat` con el nombre del archivo que corresponda.

CONFIGURACIÓN DE UN SERVIDOR NIS SECUNDARIO

Conforme crece su sitio, sin duda encontrará la necesidad de distribuir la carga del servidor NIS entre varios anfitriones. NIS ofrece soporte para ello mediante el uso de servidores NIS secundarios. Estos servidores no requieren mantenimiento adicional secundario una vez configurados debido a que el servidor NIS maestro les envía actualizaciones cuando usted reconstruye los mapas (con el comando `make`, como describimos en la sección “Edición del archivo Makefile” en este mismo capítulo).

Hay tres pasos para poner en marcha un servidor NIS secundario:

1. Configuración del nombre de dominio NIS.
2. Configuración del NIS maestro para la propagación hacia servidores esclavos.
3. Activación del servidor esclavo con `ypinit`.

Configuración del nombre de dominio

Así como lo hizo cuando configuró un servidor NIS maestro, debería establecer el nombre del dominio NIS antes de empezar el proceso de inicio de un servidor secundario (servidorB):

```
[root@servidorB ~]# domainname mi_nombre_dominio
```

donde `mi_nombre_dominio` es el nombre de dominio NIS para su sitio.

Desde luego, el nombre de dominio del servidor secundario debe configurarse de manera que éste se establezca en el momento de arranque. Si está utilizando la versión Fedora Core de Linux, como en nuestro sistema de muestra, establezca la variable `NISDOMAIN` en el archivo `/etc/sysconfig/network`. De lo contrario, edite el archivo `/etc/init.d/ypserv` de manera que lo primero que aparezca, después de los comentarios iniciales, sea la configuración del nombre de dominio.

NOTA Asegúrese de establecer el nombre de dominio a mano antes de continuar con el paso **ypinit** de instalación.

Configuración del NIS maestro para la propagación hacia servidores esclavos

Si todavía no ha configurado el servidor NIS maestro que hará la propagación hacia los servidores NIS esclavos, deberá hacerlo ahora. Ello requiere de dos pasos: primero edite el archivo **/var/yp/ypservers** a fin de que éste incluya todos los servidores NIS secundarios hacia los cuales el servidor NIS maestro propagará los mapas. Por ejemplo, si quiere que el servidor maestro propague mapas hacia los anfitriones servidorB y servidorC, editaría el archivo **/var/yp/ypservers** de manera que su contenido se vea como sigue:

```
servidorA  
servidorB  
servidorC
```

donde servidorA es el nombre del anfitrión que será el servidor NIS maestro.

En segundo lugar, necesitará asegurarse de que el archivo **Makefile** incluye la línea **NOPUSH=false**. Vea la sección sobre configuración de servidores NIS maestros para consultar más detalles.

Activación de **ypinit**

Una vez efectuados los pasos anteriores, está listo para ejecutar el comando **ypinit** para iniciar el servidor secundario. Escriba el siguiente comando en el servidor NIS secundario:

```
[root@servidorB ~]# /usr/lib/yp/ypinit -s servidorA
```

donde la opción **-s** ordena a **ypinit** que configure el sistema como un servidor esclavo, y **servidorA** es el nombre del servidor NIS maestro.

La respuesta de este comando le informará que **ypxfrd** no se está ejecutando; puede ignorar este mensaje. Lo que el servidor secundario está tratando de hacer es extraer los mapas del servidor NIS maestro utilizando el demonio **ypxfrd** para ello. Esto no funcionará porque no configuró al servidor NIS maestro para aceptar solicitudes de extracción de mapas mediante **ypxfrd**. En vez de ello, configuró al servidor maestro para propagar mapas hacia los servidores secundarios cuando el maestro tuviera una actualización. En este punto, el proceso del servidor debe iniciarse a mano con el mismo proceso que utilizó para el servidor primario: **ypserv**. Para iniciar lo, ejecute el siguiente comando:

```
[root@servidorB ~]# service ypserv start
```

NOTA Asegúrese de hacer que el proceso del servidor inicie como parte del proceso de arranque de la máquina. Puede utilizar el programa **chkconfig** para lograrlo. El programa **ypserv** debe iniciar en los niveles de ejecución 3 y 5.

Para probar el servidor secundario vaya al servidor maestro e intente hacer una propagación provocada por el servidor. Puede lograrlo si ejecuta otra vez el programa **make** en el servidor NIS maestro, como sigue:

```
[root@servidorA ~]# cd /var/yp ; make
Updating passwd.byname...
Updating passwd.byuid...
Updating group.byname...
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
Updating mail.aliases...
```

Esto deberá forzar la regeneración de los mapas y su propagación desde el servidor maestro hacia los servidores secundarios.

HERRAMIENTAS NIS

A fin de ayudarle a trabajar con NIS, un puñado de herramientas le permite extraer información de la base de datos en modo texto, desde una línea de comandos.

- ▼ **ypcat**
- **ypwhich**
- **ypmatch**
- ▲ **ypasswd**

La primera herramienta, **ypcat**, vacía el contenido de un mapa NIS. Ello es útil para scripts que necesitan extraer información de NIS: **ypcat** puede obtener mapas enteros y luego **grep** puede usarse para encontrar una entrada específica. El comando **ypcat** también es útil para realizar pruebas sencillas al servicio NIS. Por ejemplo, para utilizar **ypcat** (y **grep**) con el fin de vaciar y encontrar una entrada para el usuario yyang en la base de datos passwd, escriba:

```
[root@servidorA ~]# ypcat passwd | grep yyang
yyang:$1$ cXSafue2DYg0zskw.Hj/:500:500:Ying Yang:/home/yyang:/bin/bash
```

El comando **ypwhich** regresa el nombre del servidor NIS que está respondiendo a sus solicitudes. Ello también es una buena herramienta de diagnóstico si NIS no parece estar trabajando como se espera. Por ejemplo, supongamos que ha hecho un cambio en las tablas del servidor NIS maestro pero sus cambios no pueden ser vistos por un cliente específico. Puede utilizar **ypwhich** para ver a cuál servidor está ligado el cliente. Si está ligado a un servidor secundario, quizás se deba a que el servidor secundario no está listado en el archivo **/var/yp/ypservers** del servidor primario.

Enseguida mostramos un ejemplo de uso de **ypwhich**:

```
[root@servidorA ~]# ypwhich
```

El comando **ypmatch** es un pariente cercano de **ypcat**. Sin embargo, en vez de obtener un mapa en su totalidad, si usted proporciona un valor llave a **ypmatch**, sólo trae la entrada que

corresponda a ese valor. Utilizando como ejemplo el mapa **passwd** podemos extraer la entrada que corresponde al usuario yyang con este sencillo comando:

```
[root@servidorA ~]# ypmatch yyang passwd
```

El comando **yppasswd** es la versión NIS del comando estándar **passwd** de Linux. La diferencia entre los dos es que el comando **yppasswd** permite al usuario establecer su contraseña en el servidor NIS. El comportamiento es idéntico a **passwd**. De hecho, varios sitios renombran el comando **passwd** a algo así como **passwd.local** y luego crean un symlink (enlace simbólico) desde **passwd** hacia **yppasswd**.

USO DE NIS EN ARCHIVOS DE CONFIGURACIÓN

Uno de los usos más populares de NIS es compartir el archivo **/etc/passwd** de manera que todos puedan registrarse en todos los anfitriones dentro de la red con sólo hacer una modificación al mapa maestro en **/etc/passwd**. Algunas distribuciones de Linux respaldan esta característica de manera automática en cuanto detectan que NIS está ejecutándose. Algunas otras todavía requieren configuraciones explícitas en el archivo **/etc/passwd** de manera que el programa de registro sepa que debe asomarse a NIS al igual que al archivo base de contraseñas.

Supongamos que necesita agregar las fichas especiales a su archivo **/etc/passwd** con el propósito de permitir registros de usuarios listados en el archivo **passwd** de NIS.

Enseguida está la configuración básica que necesitará añadir al archivo **/etc/passwd** de su cliente para permitir el registro en ese anfitrión a todos los usuarios listados en la lista **passwd** de NIS:

```
+ : * : : : :
```

NOTA Cualquier sistema basado en glibc (por ejemplo, Fedora, RHEL, Red Hat) no necesita esta adición al archivo **/etc/passwd**, aunque tenerla no confundirá a glibc ni ocasionará que se comporte en forma defectuosa.

Y enseguida está la configuración si quiere prevenir que cualquiera se registre en ese anfitrión, a excepción de aquellas personas listadas en forma explícita en el archivo **/etc/passwd**.

```
+ : : : : : /bin/false
```

Esto anula todas las configuraciones de la sesión del usuario de manera que, cuando alguien intenta registrarse en el cliente, el programa de registro intenta ejecutar **/bin/false** como el programa de registro del usuario. Como **/bin/false** no opera como un shell, el usuario es inmediatamente sacado del sistema.

Para permitir el acceso a algunos usuarios listados en forma explícita al tiempo que niega el acceso de todos los demás, utilice las entradas muestra a continuación en el archivo **/etc/passwd**:

```
+nom_usuario  
+nom_usuario2  
+nom_usuario3  
+ : : : : : /bin/false
```

En forma específica, esto sólo permite el acceso al sistema a los usuarios **nom_usuario**, **nom_usuario2** y **nom_usuario3**.

PUESTA EN MARCHA DE NIS EN UNA RED EXISTENTE

En esta sección abordaremos la implementación de NIS en ambientes de red de la vida real. Esto no intenta ser un libro de recetas sino una colección de ejemplos. Después de todo, ya vimos los detalles de la configuración y puesta en marcha de servidores y clientes de NIS. ¡No lo repetiremos!

Es obvio que habrá excepciones para cada uno de los escenarios planteados aquí. Algunas redes pequeñas generarán una considerable cantidad de tráfico NIS, por alguna razón particular. Por otro lado, algunas redes grandes tendrán un tráfico NIS tan ligero que será suficiente con un solo servidor maestro. En cualquier caso, aplique una dosis generosa de sentido común a las siguientes secciones y no tendrá mayor problema.

Una red pequeña

Consideramos que una red es pequeña cuando tiene menos de 30 a 40 sistemas Linux/UNIX, todos dentro de la misma subred.

En este caso, un solo servidor NIS maestro es más que suficiente. A menos que cualquiera de sus sistemas en la red esté generando una inusual cantidad de solicitudes de NIS, los demás sistemas pueden configurarse como clientes que consultarán al servidor maestro, ya sea por el método de difusión o por el método de conexión directa. Si no planea segmentar su red, quizás quiera apegarse al uso del método de difusión debido a que éste simplifica el proceso para añadir anfitriones a la red.

El servidor NIS en sí no debería ser demasiado pesado. Si tiene la fortuna de contar con una máquina poderosa para realizar la tarea, no dude en compartir la carga con otro servidor ligero u otros dos (DHCP es por lo general un buen candidato para compartir cargas).

Una red segmentada

Las redes segmentadas introducen cierta complejidad al proceso de manejar servicios de difusión (como ARP o DHCP). Sin embargo, para una red que crece, la segmentación es ante todo una necesidad. Al segmentar el tráfico en dos o más redes discretas, podrá mantener el tráfico en cada segmento a niveles bajos, de fácil control. Además, este arreglo le permite imponer una seguridad más rigurosa para sistemas internos. Por ejemplo, puede poner a Contabilidad y Recursos Humanos en otra subred a fin de dificultar que los de Ingeniería puedan “oler” el tráfico de la red para extraer información confidencial.

Para NIS la segmentación significa dos posibles soluciones. En la primera solución se supone que, a pesar de que usted tiene una red más grande, no requiere de mucho tráfico NIS. Esta es la típica situación dentro de redes heterogéneas en la que Microsoft Windows se ha instalado en muchas de las estaciones de trabajo. En este caso, mantener un solo servidor NIS maestro es más que suficiente. *De cualquier forma*, los clientes de esta red se deben configurar para contactar al servidor de manera directa en vez de utilizar la técnica de difusión. Ello es así debido a que solamente aquellos clientes que estuvieran en la misma subred que el servidor NIS serían capaces de contactarlo mediante la técnica de difusión; y, además, es mucho más fácil mantener todas sus estaciones de trabajo configuradas de manera consistente.

Por otro lado, si considera que sí hay suficiente tráfico NIS, es una buena idea repartir la carga entre varios servidores, uno para cada subred. En este caso, el servidor NIS maestro se configura para propagar actualizaciones hacia los servidores secundarios cuando los mapas sean actualizados en forma local; y los clientes pueden ser configurados consistentemente para utilizar la técnica de

difusión a fin de encontrar el servidor NIS correcto. Cuando utiliza el método de difusión, los clientes pueden moverse desde una subred a otra sin que deba reasignar su servidor NIS.

Redes de mayor tamaño que los edificios

No es raro que el tamaño de las redes supere el tamaño de los edificios en los que están ubicadas. Las oficinas remotas conectadas a través de una variedad de métodos significan una variedad de decisiones administrativas, ¡las cuales no sólo conciernen a NIS!

Sin embargo, para NIS es crucial que un servidor sea ubicado en cada extremo de toda conexión WAN. Por ejemplo, si tiene tres universidades conectadas entre sí con enlaces T1, debería contar con tres servidores NIS, cuando menos, uno para cada universidad. Este arreglo es necesario debido a que NIS confía en la baja latencia de los enlaces para su buen desempeño, máxime si consideramos que es un protocolo basado en RPC (la ejecución de un simple comando `1s -1` puede resultar literalmente en cientos de búsquedas). Además, en caso de que uno de los enlaces WAN falle, es importante que cada sitio sea capaz de operar por su cuenta hasta que el enlace sea reestablecido.

La existencia de varios dominios NIS en su red, lo que implica su segmentación, depende de la organización de su compañía y de su administración. Una vez que tome esta decisión administrativa podrá tratar a cada universidad como un sitio independiente y entonces decidir cuántos servidores NIS necesitará. Si su intención es mantener un espacio NIS, debería contar con un solo servidor NIS maestro; el resto de los servidores NIS en las otras universidades deberían ser esclavos.

RESUMEN

En este capítulo discutimos el proceso de instalar servidores NIS maestros, servidores NIS esclavos y clientes de NIS. Asimismo, vimos cómo utilizar algunas de las herramientas disponibles en estos servidores. Enseguida mostramos los puntos clave que hay que recordar acerca de NIS:

- ▼ Aunque de naturaleza similar a los controladores de dominio Windows, los servidores NIS no son iguales. Un aspecto en particular es que los servidores NIS no llevan a cabo autenticación.
- Debido a que cualquier persona en su red puede unirse a un dominio NIS, se supone que su red ya es segura. Muchos sitios encuentran que los beneficios de este arreglo exceden los riesgos.
- Una vez que se configura el archivo **Makefile** y **ypinit** inició la ejecución, los servidores NIS maestros no necesitan de configuraciones adicionales. Los cambios a los archivos que necesita compartir mediante NIS (como el archivo **/etc/passwd**) se actualizan y propagan con el uso del comando **cd /var/yp; make**.
- Los servidores NIS esclavos se listan en el archivo **ypinit**, mismo que reside en el servidor maestro.
- Los servidores NIS esclavos reciben las actualizaciones desde el servidor maestro mediante un proceso conocido como *propagación hacia servidores* (*server push*).
- A efecto de poner en marcha un servidor NIS esclavo, son necesarias más tareas que la sola ejecución del comando **ypinit -s**.

- Los clientes de NIS necesitan que los archivos `/etc/yp.conf` y `/etc/nsswitch.conf` se configuren en forma apropiada.
- ▲ Cuando una distribución Linux en particular lo requiera, asegúrese de hacer ajustes finos a NIS en el archivo de contraseñas que reside del lado del cliente. La mayoría de los sistemas basados en Red Hat no requieren estos ajustes.

CAPÍTULO 24



Samba

Samba es una poderosa herramienta que permite a los sistemas basados en UNIX (como Linux) la posibilidad de interactuar con sistemas que se basan en Windows y otros sistemas operativos. Es la implementación de fuente abierta de un par de protocolos llamados Server Message Block (SMB) (Bloque de mensajes del servidor) y Common Internet File System (CIFS) (Sistema común de archivos para Internet).

Samba proporciona servicios transparentes para compartir archivos e impresoras entre clientes de Windows. Ello es posible gracias al uso de SMB/CIFS, dos protocolos de red procedentes de Microsoft. Desde el punto de vista de un administrador de sistemas, esto significa contar con la capacidad de poner en marcha un servidor basado en UNIX, sin la necesidad de instalar NFS, LP y algún tipo de soporte de autenticación compatible con UNIX en todos los clientes de Windows en una red. En vez de ello, los clientes pueden utilizar su dialecto nativo para comunicarse con el servidor, lo que significa menos problemas para usted y una integración sin asperezas para sus usuarios.

Este capítulo cubre el procedimiento de descarga, compilación e instalación de Samba. Por fortuna, las configuraciones predeterminadas de este software requieren de pocas modificaciones, así que nos concentraremos en cómo llevar a cabo tareas personalizadas y en cómo evitar algunos de los tropiezos más comunes. En lo que se refiere a la administración, le daremos un pequeño curso sobre el uso de la Samba's Web Administration Tool (SWAT) (Herramienta Web para administración de Samba), y sobre la herramienta para la línea de comandos **smbclient**.

No importa qué tarea haya escogido para Samba, asegúrese de que se toma un tiempo para leer la documentación del programa. Está bien escrita, es completa y exhaustiva. El medio día que pueda dedicar a su lectura le brindará una valiosa cantidad de conocimientos.

NOTA En realidad Samba ha sido migrada hacia un número significativo de plataformas, casi para cualquier variante de UNIX que usted pueda imaginar, incluso para varios ambientes ajenos a UNIX. En este análisis, desde luego, estamos más interesados en Samba/Linux, pero tenga presente que también lo puede poner en marcha en sus otros sistemas de UNIX.

LA MECÁNICA DE SMB

Para comprender en su totalidad la relación Linux/Samba/Windows necesita entender las relaciones entre el sistema operativo y sus archivos, impresoras, usuarios y redes. Para ver con mayor detalle cómo se confrontan estas relaciones, examinemos algunos de los aspectos fundamentales al trabajar con Linux y Windows en un mismo ambiente.

Nombres de usuarios y contraseñas

El mecanismo para nombre de usuario/contraseña en Linux/UNIX es radicalmente distinto del modelo de Primary Domain Controller (PDC) (Controlador de dominio primario) de Windows y del modelo Active Directory de Windows 2000. Así que para el administrador es importante mantener consistencia en los registros y contraseñas de ambos sistemas. Los usuarios necesitan acceder a ambos sistemas sin tener que preocuparse por la autenticación reiterada o que las contraseñas almacenadas en caché no empaten con algún servidor en particular.

Tiene varias opciones de gestión para manejar esquemas de nombres de usuario y contraseñas:

- ▼ **Módulos de enchufables de autenticación (PAM, Pluggable Authentication Modules) de Linux** Esta opción le permite autenticar usuarios contra un PDC. Ello significa que todavía tendrá dos listas de usuarios que mantener, una local y una en el PDC, pero sus usuarios sólo tendrán que poner atención a las contraseñas en el sistema Windows.
- **Samba como un PDC** Esta segunda opción le permite mantener todos sus nombres de usuario y contraseñas en el sistema Linux mientras que todas sus máquinas Windows realizan la autenticación con Samba. Cuando Samba se usa para ello en conjunto con un LDAP de segundo plano, tendrá una solución muy escalable y ampliable.
- ▲ **Su propia solución desarrollada en Perl** Esta tercera opción le permite utilizar sus propios scripts. Para sitios que cuentan con un sistema bien establecido para mantenimiento de registros de usuario y contraseñas no resulta irrazonable utilizar scripts a la medida. Ello puede lograrse utilizando WinPerl y módulos Perl, que permiten cambios al Security Access Manager (SAM) (Administrador de seguridad de accesos), con el fin de actualizar la lista de contraseñas del PDC. Un script Perl del lado de Linux puede comunicarse con un script WinPerl a fin de mantener las cuentas sincronizadas.

En el peor de los escenarios, puede mantener los dos sistemas *a mano* (lo que algunos sysadmins de la vieja escuela en realidad hacían!), pero este método es propenso a errores y no es muy divertido.

Contraseñas con encriptación

Desde la aparición de Windows NT4/Service Pack 3, Windows 98 y Windows 95 OSR2, el sistema operativo Windows utiliza contraseñas con encriptación cuando se comunica con el PDC y cualquier servidor que requiera autenticación (incluyendo Linux y Samba). Sin embargo, el algoritmo de encriptación utilizado por Windows es diferente del algoritmo utilizado en UNIX y, por lo tanto, no son compatibles.

Estas son sus opciones para enfrentar este conflicto:

- ▼ Edite el Registro de Windows en los clientes para inhabilitar el uso de contraseñas con encriptación. Las entradas del Registro que necesita cambiar se listan en el directorio `docs` de Samba. Desde la versión 3 de Samba, esta opción ya no es necesaria.
- ▲ Configure Samba para que utilice contraseñas con encriptación estilo Windows.

La primera solución tiene la ventaja de no obligarlo a utilizar un esquema de contraseñas más complejo. La desventaja es que tiene que efectuar las modificaciones al Registro de Windows en todos sus clientes. La segunda opción, desde luego, tiene el efecto opuesto: a cambio de un esquema ligeramente más complejo, no tiene que realizar modificaciones en los clientes.

Demonio Samba

El servidor Samba en realidad se compone de tres demonios: **smbd**, **nmbd** y **winbindd**.

El demonio **smbd** es el que maneja recursos compartidos como sistemas de archivos y servicios de impresión para clientes. También es responsable de la autenticación de usuarios y cuestiones de cierre de recursos. Inicia su operación enlazándose con el puerto 139 y con el puerto 445 para escuchar solicitudes. Cada vez que un cliente se autentifica, **smbd** crea una copia de sí

mismo; el original regresa a escuchar las nuevas solicitudes que llegan por el puerto primario y la copia maneja la conexión con el cliente. Además, la nueva copia cambia su ID de raíz a aquella del usuario autenticado (por ejemplo, si el usuario yyang se autentica contra **smbd**, la nueva copia operaría con los permisos de yyang, no con los permisos de raíz). La copia permanece en la memoria en tanto exista la conexión desde el cliente.

El demonio **nmbd** es responsable de manejar las solicitudes del servicio de nombres NetBIOS. **nmbd** también se puede usar como un reemplazo directo de un Windows Internet Name Server (WINS) (Servidor Windows de nombres de Internet). Inicia su operación enlazándose con el puerto 137; sin embargo, a diferencia de **smbd**, **nmbd** no crea una nueva instancia de sí mismo para atender cada consulta. Además de las solicitudes para el servicio de nombres, **nmbd** también atiende solicitudes que provienen de navegadores maestros, navegadores de dominio y servidores WINS, y como tal, juega un papel en los protocolos de navegación que forman parte del popular “Entorno de red” de Windows (Windows Network Neighborhood). Los servicios ofrecidos por los demonios **smbd** y **nmbd** son complementarios.

Por último, el servicio que proporciona **winbindd** se puede usar para consultar información de usuarios y grupos en servidores Windows; esta información también se puede usar en plataformas Linux/UNIX. Este programa logra su cometido empleando llamadas RPC de Microsoft, PAM, y las funciones del Name Service Switch (NSS) (Comutador del servicio de nombres) que están disponibles en las modernas bibliotecas de C. Su uso se puede ampliar mediante el empleo de un módulo PAM (**pam_winbind**) a fin de que pueda proporcionar servicios de autenticación. Este servicio se controla aparte del servicio **smb** principal y además puede funcionar en forma independiente.

NOTA Con el lanzamiento de Windows 2000, Microsoft incorporó el uso de convencionalismos de nombres DNS como parte de su respaldo a Active Directory en un intento por hacer que los servicios de nombres fueran más consistentes entre el “Entorno de red” y los nombres de anfitrión publicados en DNS. En teoría, ya no debería necesitar **nmbd**, pero la realidad es que sí lo necesitará, especialmente si planea permitir que los anfitriones no basados en Windows 2000 accedan a los recursos compartidos con Samba.

Instalación de Samba

En la mayoría de las distribuciones de Linux vienen incluidos binarios precompilados de Samba. Esta sección le mostrará cómo instalar este software mediante RPM en un sistema Fedora Core. Necesita dos paquetes en sistemas Fedora Core y RHEL a fin de proveer los servicios de Samba que van del lado del servidor. Éstos son

- ▼ **samba*.rpm** Este paquete ofrece la funcionalidad de un servidor SMB que se puede usar para proporcionar servicios a los clientes de SMB/CIFS.
- ▲ **samba-common*.rpm** Este paquete proporciona los archivos necesarios para los paquetes del servidor y del cliente, archivos tales como los archivos de configuración, archivos de registro, manuales, módulos PAM y otras bibliotecas.

Se requiere otro paquete a fin de que los sistemas Fedora Core y RHEL puedan proporcionar los servicios Samba que van del lado del cliente. Dicho paquete es:

- ▼ **samba-client*.rpm** Este paquete proporciona las utilidades del cliente que permiten el acceso a los recursos compartidos SMB y a los servicios de impresión en sistemas Linux y en otros sistemas.

Suponiendo que tiene una conexión funcional a Internet, la instalación de Samba puede ser tan sencilla como emitir el siguiente comando:

```
[root@servidorA ~]# yum install samba
```

De igual forma, puede instalar el paquete **samba-client** como sigue:

```
[root@servidorA ~]# yum install samba-client
```

Si lo prefiere, puede realizar la descarga e instalación del paquete RPM desde el almacén de software del distribuidor (<http://fedora.redhat.com/download/>) o desde el directorio **/Fedora/RPMS/** del medio de instalación utilizando los comandos RPM de costumbre, por ejemplo,

```
[root@servidorA ~]# rpm -ivh /media/cdrom/Fedora/RPMS/samba-*.rpm
```

Compilación e instalación de Samba desde el código fuente

En la mayoría de las distribuciones de Linux, Samba ya viene instalado. No obstante, como sucede con todos los servicios que hemos discutido en esta obra, debe serle posible compilar el software por su cuenta en caso de que quiera asegurarse que instala la última versión. Desde su aparición Samba ha tenido usuarios en muchas de las plataformas UNIX/Linux, razón por lo que ha sido diseñada para ser compatible con todas las variantes. Es muy raro encontrar problemas durante el proceso de compilación.

Al momento de escribir esta obra, la última versión confiable del código fuente de Samba es la 3.0.14a.

Empiece por descargar el código fuente de Samba desde <http://www.samba.org> y guárdelo en el directorio donde quiera compilarlo. Para este ejemplo supondremos que ese directorio es **/usr/local/src**. También puede descargar la última versión directamente desde <http://us4.samba.org/samba/ftp/samba-latest.tar.gz>.

1. Desempaque Samba utilizando el comando **tar**.

```
[root@servidorA src]# tar xvzf samba-latest.tar.gz
```

2. El paso 1 crea un subdirectorio llamado **samba-3.0.14a** para el código fuente. Cámbiese a ese directorio. Escriba

```
[root@servidorA src]# cd samba-3.0.14a/
```

SUGERENCIA Con su editor de texto favorito empiece por leer el archivo **Manifest**, el cual explica todos los archivos que vienen con Samba y le dice dónde se encuentra su documentación. Aunque de entrada ello no es crucial, le ayudará más adelante.

3. Dentro del directorio **samba-3.0.14a**, habrá otro directorio llamado **source**. Entre a ese directorio como sigue:

```
[root@servidorA samba-3.0.14a]# cd source/
```

SUGERENCIA Es posible que el script de configuraciones no esté en el directorio **source** de Samba. Puede confirmarlo listando los archivos en la carpeta si utiliza el comando **ls**. Si dicho script no está presente, tendrá que crearlo mediante el script **autogen.sh** que se encuentra dentro del directorio fuente del árbol fuente de Samba.

- Habilitaremos el soporte para **smbmount** con el script de configuraciones de Samba. Las otras opciones que quizás quiera considerar están listadas en la tabla 24-1. En este ejemplo, la opción **smbmount** será la única que habilitaremos, lo que implica aceptar el resto de los valores predeterminados. Escriba

```
[root@servidorA source]# ./configure --with-smbmount
```

- Empiece por compilar Samba mediante el uso del comando **make**.

```
[root@servidorA source]# make
```

- Ahora utilice **make install**.

```
[root@servidorA source]# make install
```

- Está listo. Encontrará todos los binarios de Samba y los archivos de configuración instalados dentro del directorio **/usr/local/samba/**. Ahora puede continuar y utilizar los archivos tal como si los hubiera instalado con RPM. Desde luego, ¡deberá poner atención en las rutas de los archivos!

Opciones de configuración de Samba (./configure)	Descripción
--prefix=PREFIX	Instala los archivos de arquitectura independiente en PREFIX.
--with-smbmount	Incluye soporte para el comando smbmount . Este comando permite adjuntar recursos compartidos desde servidores NT (u otros servidores Samba) casi de la misma forma como se montan particiones NFS.
--with-pam	Incluye soporte para los PAM (valor predeterminado=no).
--with-ldapsam	Incluye una configuración compatible con LDAP SAM 2.2 (valor predeterminado=no).
--with-ads	Soporte para Active Directory (valor predeterminado=auto).
--with-ldap	Soporte para LDAP (valor predeterminado=yes).
--with-pam_smbpass	Construye el módulo PAM para autenticación contra bases de datos passdb de segundo plano.
--with-krb5-base-dir	Ubica el soporte Kerberos 5 (valor predeterminado=/usr).
--enable-cups	Habilita el soporte para CUPS (valor predeterminado=auto).

Tabla 24-1. Algunas de las opciones de configuración (**./configure**) de Samba

NOTA Es usual que el directorio **/usr/local/samba/bin** no se encuentre en la ruta de búsqueda para la mayoría de los shells. Puede añadirlo a la ruta de búsqueda o simplemente copiar los binarios desde **/usr/local/samba/bin** hacia una ubicación donde se puedan encontrar (por ejemplo, **/usr/sbin** o **/usr/bin**).

ADMINISTRACIÓN DE SAMBA

Esta sección describe algunas consideraciones sobre la administración de Samba. Veremos cómo iniciar y detener Samba, cómo realizar tareas comunes de administración con SWAT y cómo utilizar **smbclient**. Por último examinaremos el proceso para utilizar contraseñas con encriptación.

Inicio y terminación de Samba

Casi todas las distribuciones de Linux tienen scripts y programas que iniciarán y detendrán Samba sin tener que hacer nada más en especial. Dichas herramientas se encargan de iniciar los servicios durante el arranque y también de detenerlos durante el apagado. En nuestro sistema de muestra, el cual ejecuta Fedora con Samba instalado mediante RPM, el comando **service** y la utilidad **chkconfig** pueden utilizarse para administrar el inicio y la terminación de Samba.

Por ejemplo, para iniciar el demonio **smbd**, puede ejecutar el siguiente comando:

```
[root@servidorA ~]# service smbd start
```

Y para detener el servicio, escriba

```
[root@servidorA ~]# service smbd stop
```

Después de hacer cualquier cambio en la configuración de Samba deberá reiniciar los servicios con el siguiente comando para hacer que los cambios surtan efecto:

```
[root@servidorA ~]# service smbd restart
```

El servicio **smb** en Fedora Core no iniciará en forma automática en el siguiente reinicio del sistema. Puede configurarlo para que lo haga con la utilidad **chkconfig** como sigue:

```
[root@servidorA ~]# chkconfig smb on
```

SUGERENCIA Iniciar el Samba que instalamos a partir del código fuente puede hacerse con el siguiente comando:

```
[root@servidorA ~]# /usr/local/samba/sbin/smbd -D
```

El único parámetro de línea de comando utilizado aquí (**-D**) ordena a **smbd** que funcione como demonio. De igual forma, el demonio **nmbd** puede iniciarse con

```
[root@servidorA ~]# /usr/local/samba/sbin/nmbd -D
```

Resulta un poco más difícil detener a Samba sin el uso de los scripts apropiados. Quizá tenga que utilizar el comando **ps** para listar todos los procesos de Samba. De la lista resultante encuentre la instancia de **smbd** cuyo propietario es raíz y mate el proceso. Esto también matará el resto de las conexiones de Samba.

USO DE SWAT

SWAT es la Herramienta Web para la Administración de Samba con la cual puede administrar este software desde la interfase de un navegador. Es una excelente alternativa para evitar la edición manual de los archivos de configuración (**smb.conf** y demás).

Antes de la versión 2.0 de Samba, la manera oficial de configurar este software era editar el archivo **smb.conf**. Aunque de naturaleza comunicativa y de fácil entendimiento, lidiar con este archivo era algo incómodo debido a sus numerosas opciones y directrices. También significaba que era mucho más fácil compartir recursos desde MS Windows que compartirlos con Samba. Muchos individuos crearon interfaces gráficas para el proceso de edición. Varias de estas herramientas aún se mantienen y mejoran; puede leer más sobre ellas si visita el sitio Web de Samba en <http://www.samba.org>. Sin embargo, desde la versión 2.0, el código fuente se acompaña de SWAT, la Herramienta Web para la Administración de Samba.

El software SWAT es empacado como un paquete separado en sistemas Fedora Core y RHEL. El binario RPM que provee SWAT se llama **samba-swat**. En esta sección instalaremos este paquete con el programa Yum.

Puesta en marcha de SWAT

Lo que hace a SWAT ligeramente diferente de otras herramientas de administración basadas en Web es que SWAT no depende de un servidor Web adicional (como Apache). En vez de ello, SWAT lleva a cabo todas las funciones Web requeridas sin implementar un servidor Web completo.

La puesta en marcha de SWAT es un proceso bastante directo. Éstos son los pasos:

1. Utilice Yum para descargar e instalar SWAT. Escriba

```
[root@servidorA ~]# yum install samba-swat
```

Escriba **yes** para confirmar la instalación de **samba-swat** y cualquiera de sus dependencias cuando se le pregunte.

SUGERENCIA La herramienta SWAT es empacada con el principal árbol fuente de Samba, así que se compila cuando construye Samba desde el código fuente. En la compilación y construcción de Samba que vimos páginas atrás, SWAT se instaló dentro del directorio **/usr/local/samba/swat/directory**.

2. Confirme que ya tiene instalado el paquete **samba-swat**. Escriba

```
[root@servidorA ~]# rpm -q samba-swat  
samba-swat-3.0.14a-2
```

3. SWAT funciona bajo el control del superdemonio **xinetd**. SWAT está desactivado en forma predeterminada. Corrobore su estado escribiendo

```
[root@servidorA ~]# chkconfig --list swat  
swat           off
```

4. Active SWAT escribiendo la siguiente instrucción

```
[root@servidorA ~]# chkconfig swat on
```

5. Reinicie **xinetd** para que los cambios surtan efecto. Escriba

```
[root@servidorA ~]# service xinetd restart
Stopping xinetd:                                     [FAILED]
Starting xinetd:                                     [  OK  ]
```

6. Por último, puede conectarse a la interfaz Web de SWAT utilizando un navegador Web en el mismo sistema donde esté instalado. Lleve su navegador Web hacia el URL de SWAT,

`http://localhost:901/`

Al entrar a este URL se le preguntará un nombre de usuario y una contraseña con la cual habrá de registrarse en SWAT. Escriba **root** como el nombre del usuario y luego la contraseña de raíz (root). Una vez que su registro se complete con éxito, se mostrará una página Web similar a la de la figura 24-1.

Y eso es prácticamente todo lo que debe hacer para instalar y habilitar SWAT en un sistema Fedora.

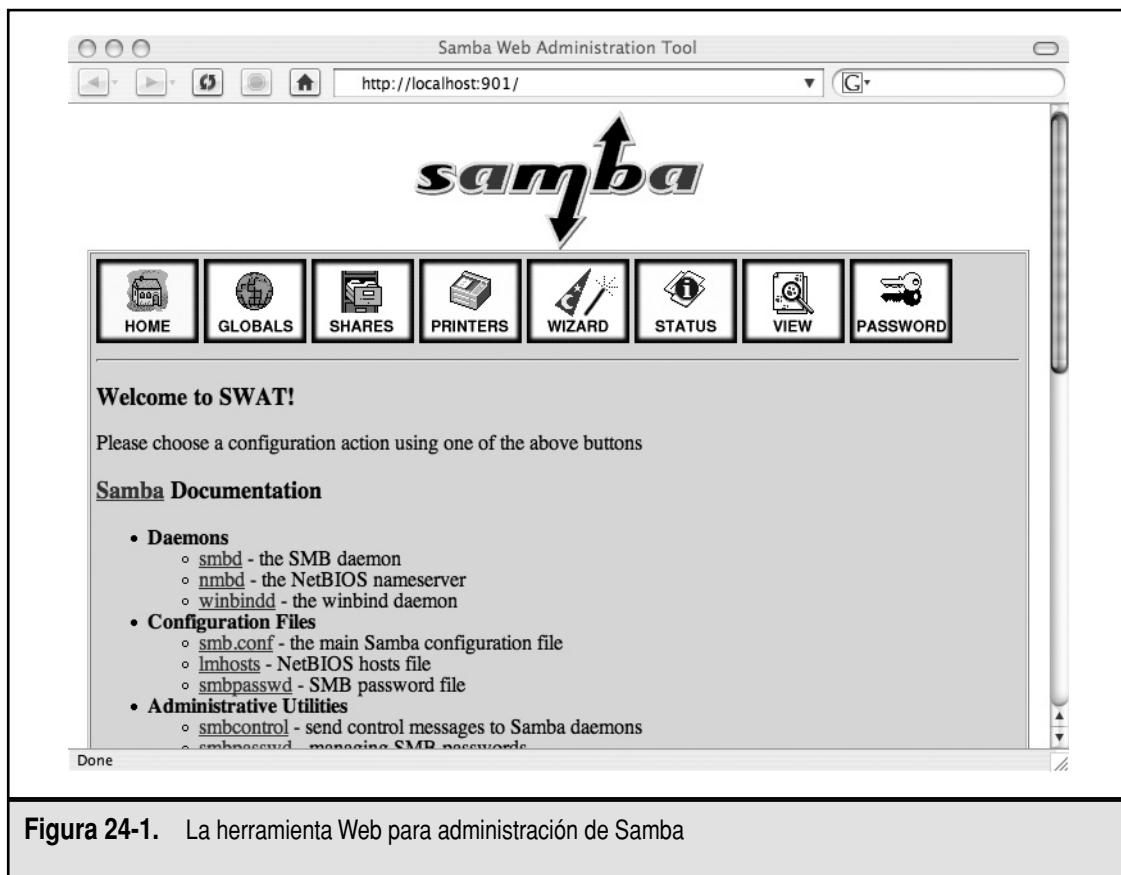


Figura 24-1. La herramienta Web para administración de Samba

NOTA La configuración predeterminada de SWAT para **xinetd** permite conectarse a SWAT sólo desde la misma máquina en la cual está ejecutando Samba (esto es, el anfitrión local). Ello se debe a motivos de seguridad: no querría que otras personas pudieran conectarse en forma remota a su servidor para “ayudarle” a configurarlo, ¿o sí?

CUIDADO Registrarse en SWAT como el usuario raíz ocasiona que su contraseña sea enviada desde el navegador Web hacia el servidor Samba. Por lo tanto, evite realizar tareas administrativas a través de una red no confiable. De preferencia, conéctese sólo desde el mismo servidor, o ponga en marcha un túnel SSH entre el anfitrión cliente y el servidor Samba.

Menús de SWAT

Cuando se conecta a SWAT y se registra como el usuario raíz verá el menú principal mostrado en la figura 24-1. Desde aquí podrá encontrar casi toda la documentación que necesitará sobre los archivos de configuración, los demonios y los programas relacionados. Ninguno de los vínculos apunta a sitios Web externos, así que puede leerlos a sus anchas sin necesidad de conectarse a la red.

En la parte superior de la página principal de SWAT están botones para elegir alguna de las siguientes opciones:

Home	La página del menú principal
Globals	Opciones de configuración que afectan todos los aspectos operacionales de Samba
Shares	Para configurar recursos compartidos en discos y sus respectivas opciones
Printers	Para la configuración de impresoras compartidas
Status	El estado de los procesos smbd y nmbd , incluyendo una lista de todos los clientes conectados a estos procesos y lo que están haciendo (la misma información que se lista con el comando smbstatus desde un intérprete de comandos)
View	El archivo smb.conf resultante
Password	Configuraciones de contraseñas

Globals

La página Globals lista todas las configuraciones que afectan aquellos aspectos de la operación de Samba. Estos parámetros se dividen en cinco grupos: base, security, logging, browse y WINS. A la izquierda de cada opción hay un vínculo hacia la documentación relevante del parámetro y sus valores.

Shares

En MS Windows compartir un recurso puede ser tan sencillo como seleccionar una carpeta (o crearla), hacer clic en el lado derecho del ratón y permitir que sea un recurso compartido. Puede establecer controles adicionales si hace clic en el lado derecho del ratón en la carpeta y selecciona Propiedades.

Mediante el uso de SWAT estas acciones se logran generando un nuevo recurso compartido. Entonces puede seleccionar dicho recurso y hacer clic en Choose Share. Esto muestra los parámetros relacionados para el recurso compartido.

Printers

La página Printers para SWAT permite configurar parámetros relacionados con Samba para impresoras que están disponibles en el sistema. Mediante varios menús puede añadir impresoras compartidas, eliminarlas, modificarlas, etc. Lo único que no puede hacer aquí es añadir impresoras al sistema principal, eso lo tiene que hacer por otros medios (vea el capítulo 26).

Status

La página Status muestra el estado actual de los demonios **smbd** y **nmbd**. Esta información incluye cuáles clientes están conectados y sus acciones. En forma predeterminada, la página se actualiza cada 30 segundos pero puede cambiar esa frecuencia si así lo requiere (es una opción dentro de esa misma página). Además de la información de estado, puede iniciar o detener Samba o pedirle que vuelva a cargar el archivo de configuración. Ello es necesario si hace cualquier cambio a la configuración.

View

Conforme cambia la configuración de Samba, SWAT lleva control de los cambios y se las ingenia para saber la información que necesita poner en el archivo de configuración **smb.conf**. Abra la página View y verá cómo SWAT va conformando el archivo de configuración para usted.

Password

Utilice la página Password si tiene la intención de mantener contraseñas con encriptación. Si quiere ofrecer a sus usuarios una manera de modificar sus contraseñas sin tener que registrarse en el servidor Linux, esta página le permitirá hacerlo.

NOTA Casi siempre es una buena idea restringir el acceso a sus servidores a cualquiera, excepto al personal de soporte técnico. Ello disminuye las posibilidades de que alguien cometa errores que pudieran afectar el desempeño o la estabilidad de su servidor.

Creación de un recurso compartido

Lo llevaremos por el proceso de creación de un recurso compartido dentro del directorio **/tmp** que será compartido en un servidor Samba. Primero crearemos el directorio y luego editaremos el archivo de configuración de Samba (**/etc/samba/smb.conf**) para definir ahí que ese recurso será compartido.

Ello, desde luego, puede realizarse de manera sencilla con la interfaz Web de SWAT que se instaló antes, pero no la utilizaremos aquí. SWAT es muy fácil de utilizar y es intuitivo. Pero quizás es más útil entender cómo se configura Samba de manera rigurosa, pues ello hará más fácil comprender lo que SWAT hace tras bambalinas para que luego usted pueda hacer ajustes finos a

su antojo. Además, uno nunca sabe cuándo se perderá en medio del Amazonas sin unas simpáticas herramientas de configuración GUI. Así que, comencemos:

1. Cree un directorio llamado **compartidodeprueba** dentro de la carpeta **/tmp/**. Escriba

```
[root@servidorA ~]# mkdir /tmp/compartidodeprueba
```
2. Cree algunos archivos vacíos (**ado, ido, to, so, cho**) dentro del directorio que creó. Escriba

```
[root@servidorA ~]# touch /tmp/compartidodeprueba/{ado,ido,to,so,cho}
```
3. Establezca los permisos de la carpeta **compartidodeprueba** de manera que por su contenido puedan navegar otros usuarios del sistema. Escriba

```
[root@servidorA ~]# chmod -R 755 /tmp/compartidodeprueba/*
```
4. Abra el archivo de configuración de Samba para editarlo con el editor de textos de su elección y añada al final del archivo la lista que mostramos enseguida. Por favor omita los números de línea mostrados que van del 1 al 5. Esa numeración se añade sólo para fines de legibilidad.

```
1) [samba-compartida]
2)         comment=Esta carpeta contiene documentos compartidos
3)         path=/tmp/compartidodeprueba
4)         public=yes
5)         writable=no
```

La línea 1 es el nombre del recurso compartido (o servicio, como se le llama en el dialecto de Samba). Este es el nombre que verán los clientes de SMB cuando traten de navegar por los recursos compartidos del servidor Samba.

La línea 2 no es otra cosa que un comentario descriptivo que los usuarios verán junto al recurso compartido cuando estén navegando.

La línea 3 es muy importante. Especifica la ubicación dentro del sistema de archivos donde se almacena el contenido que será compartido.

La línea 4 especifica que no se requiere de ninguna contraseña para acceder al recurso compartido (a esto se le llama conectarse al servicio en el dialecto sambesco). Los privilegios habilitados en el recurso compartido se trasladarán a los permisos de la cuenta invitada que se conecta. Si en vez de ello el valor se estableciera en “no”, entonces el recurso compartido no lo podría ver el público en general pero sí usuarios permitidos autenticados.

La línea 5, con el valor de la directriz establecido en “no”, significa que los usuarios del servicio no pueden crear ni modificar los archivos contenidos ahí.

SUGERENCIA El archivo de configuración de Samba tiene demasiadas opciones y directrices como para tratarlas todas aquí. Puede aprender más sobre otras opciones posibles leyendo el manual de **smb.conf** (**man smb.conf**).

5. Guarde sus cambios en el archivo **/etc/samba/smb.conf** y salga del editor.

Note que acabamos de aceptar el resto de los valores predeterminados en el archivo. Quizá quiera regresar y personalizar algunos de los valores ajustándolos a las condiciones que imperan en su ambiente.

Un valor que quizás considere necesario cambiar es la directriz (“workgroup”) que define al grupo de trabajo. Esta directriz controla en qué grupo de trabajo el servidor aparecerá listado cuando lo consulten o lo vean clientes de Windows en el “Entorno de red”.

Note también que la configuración predeterminada quizás contenga otras definiciones. Debe comentar (o eliminar) esas entradas si no es su intención mantenerlas.

6. Use la utilidad **testparm** para revisar la validez interna del archivo **smb.conf** (es decir, para comprobar la ausencia de errores de sintaxis). Escriba

```
[root@servidorA ~]# testparm -s | less
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
[samba-compartida]
comment = Esta carpeta contiene documentos compartidos
path = /tmp/compartidodeprueba
guest ok = Yes
```

Estudie la salida buscando la aparición de errores graves e intente corregirlos regresando a editar el archivo **smb.conf**.

Note que quizás tenga que pulsar una vez la tecla Q en su teclado para salir del nivel del comando debido a que enfiló la salida de **testparm** con el comando **less**.

7. Ahora reinicie Samba (o inicie, según sea el caso) para hacer que el software reconozca sus cambios. Escriba

```
[root@servidorA ~]# service smb restart
Shutting down SMB services: [ OK ]
Shutting down NMB services: [ OK ]
Starting SMB services: [ OK ]
Starting NMB services: [ OK ]
```

Hemos terminado de crear nuestro recurso compartido. En la siguiente sección intentaremos entrar a éste.

USO DE SMBCLIENT

El programa **smbclient** es una herramienta de la línea de comandos que permite que su sistema Linux actúe como un cliente de Windows. Puede utilizar esta utilidad para conectarse a otros servidores Samba o incluso a servidores reales Windows NT/200x. **smbclient** es un programa muy flexible y se puede usar para navegar en otros servidores, enviar y recibir archivos de ellos, o incluso imprimir en ellos. Como podrá imaginar, ésta es también una excelente herramienta para depuración de errores debido a que puede verificar con rapidez y facilidad si la instalación de un nuevo servidor Samba trabaja de manera apropiada sin necesidad de utilizar un cliente de Windows para efectuar las pruebas.

En esta sección le mostraremos cómo puede realizar una navegación básica, el acceso a archivos remotos y el acceso a impresoras remotas con **smbclient**. Sin embargo, recuerde que el programa **smbclient** es un programa muy flexible, cuyo límite es sólo la imaginación de usted.

Navegación por un servidor

Con tantas interfaces gráficas en el mercado, hemos llegado a pensar que navegar es “apuntar, hacer clic y ya”. Pero cuando sus intenciones sólo requieren de encontrar lo que un servidor tiene que ofrecer, no hay razón para que recurra a una GUI completa.

Mediante el uso de **smbclient** y la opción **-L** puede ver lo que ofrece un servidor de archivos Windows o un servidor Samba, sin utilizar una GUI. Enseguida mostramos el formato de este comando:

```
[root@servidorA ~]# smbclient -L nombreanfitrío
```

donde **nombreanfitrío** es el nombre del servidor. Por ejemplo, si queremos ver lo que un anfitrión local tiene que ofrecer (esto es, servidorA), escribimos

```
[root@servidorA ~]# smbclient -L localhost
```

Quizá se le pregunte la contraseña. Puede pulsar ENTER para completar el comando.

Para listar el contenido en el servidor Samba sin que se le pregunte una contraseña, puede utilizar la opción **-U%**. Ello implica que usted quiere ser autenticado como el usuario guest (el invitado), mismo que no requiere de una contraseña. Escriba

```
[root@servidorA ~]# smbclient -U% -L localhost
```

```
Domain= [MYGROUP] OS=[Unix] Server= [Samba 3.0.14a-2]
```

Sharename	Type	Comment
-----	----	-----
samba-compartida	Disk	Esta carpeta contiene documentos compartidos
IPC\$	IPC	IPC Service (Samba Server)
ADMIN\$	IPC	IPC Service (Samba Server)

```
Domain= [MYGROUP] OS=[Unix] Server= [Samba 3.0.14a-2]
```

Server	Comment
-----	-----
SERVIDORA	Samba Server
Workgroup	Master
-----	-----
MYGROUP	SERVIDORA

Note la presencia del recurso compartido que creamos en páginas anteriores; está en la cuarta línea de la salida que acabamos de mostrar.

Acceso a archivos remotos

La utilidad **smbclient** permite acceder a archivos en un servidor Windows o en un servidor Samba mediante una interfaz de cliente híbrida DOS/FTP para la línea de comandos. Para utilizarla de la manera más clara y directa, iníciela de la siguiente forma:

```
[root@servidorB ~]# smbclient //servidor/recurso_compartido
```

donde **servidor** es el nombre del servidor (o la dirección IP) y **recurso_compartido** es el nombre del recurso compartido al cual quiere conectarse. De manera predeterminada, Samba comparte de modo automático el directorio de inicio de todos los usuarios (por ejemplo, el usuario yyang puede acceder a su directorio de inicio en el servidorA dirigiéndose a `//servidorA/yyang`).

Enseguida explicamos algunos parámetros para la línea de comandos que podrían serle de utilidad al conectarse a un servidor con **smbclient**:

Parámetro para smbclient	Descripción
-I destIP	La dirección IP destino a la cual se quiere conectar.
-U nombreusuario	El nombre del usuario con el que quiere conectarse. Este usuario se utilizará en vez del usuario con el que usted empezó su sesión.
-W nombre	Establece el nombre del grupo de trabajo a nombre .
-D directorio	Empieza la sesión conectándose al directorio .

Una vez conectado podrá navegar por los directorios utilizando los comandos **cd**, **dir** y **ls**. También puede utilizar **get**, **put**, **mget** y **mput** para transferir archivos de un lado a otro. La ayuda en línea explica todos los comandos a detalle. Podrá ver la ayuda disponible con sólo escribir **help** en el mensaje.

Hagamos un intento por establecer una conexión real al recurso compartido que antes creamos (“samba-compartida”). A fin de mostrar con mayor claridad el proceso, esto se hará desde un anfitrión distinto llamado clienteB.

Utilizaremos la utilidad **smbclient** para conectarnos al servidor, nos conectaremos como el usuario guest especificando la opción **-U%**. Después de establecer la conexión, ésta nos conducirá al shell smb con el mensaje **smb:** \>. Mientras estemos conectados listaremos los archivos en el recurso compartido con el comando **ls**. Luego intentaremos descargar uno de los archivos que residen en el recurso compartido utilizando el comando tipo FTP **get**. Por último, con el comando **quit** cerraremos la conexión. Enseguida mostramos lo que vería al efectuar una sesión como ésta, en la que se conecta el clienteB al servidorA:

```
[root@clienteB ~]# smbclient -U% //servidorA/samba-compartida
Domain= [MYGROUP] OS= [Unix] Server= [Samba 3.0.14a-2]
smb: \> ls
..
.
ado
ido
to
so
cho
D 0 Dom Mar 26 13:27:32 2006
D 0 Dom Mar 26 13:27:34 2006
A 0 Dom Mar 26 13:28:19 2006
37816 blocks of size 262144. 31717 blocks available
smb: \> get foo1
getting file \foo1 of size 0 as foo1 (0.0 kb/s) (average 0.0 kb/s)
smb: \> quit
```

El archivo (**foo1**) que se descargó desde el servidorA debería estar en el directorio actual del sistema de archivos local del clienteB.

MONTAJE DE RECURSOS REMOTOS COMPARTIDOS MEDIANTE SAMBA

Si su núcleo se configura para ofrecer soporte a sistemas de archivos SMB (como sucede con la mayoría de los núcleos que vienen en las distribuciones de Linux), podrá montar recursos compartidos de Windows o Samba en su sistema local casi de la misma forma en la que montaría una exportación NFS o un volumen local. Ello es especialmente útil para acceder a un disco grande en un servidor remoto sin tener que hacer malabares con archivos individuales a lo largo y ancho de una red.

Mientras está registrado en clienteB puede utilizar el comando **mount** con las opciones apropiadas para montar un recurso compartido Samba que reside en el servidorA.

Primero cree el punto de montaje si todavía no existe. Escriba

```
[root@clienteB ~]# mkdir -p /mnt/smb
```

Ahora ejecute el comando que hará el verdadero montaje:

```
[root@clienteB ~]# mount -t smbfs //servidorA/samba-compartida /mnt/smb
```

donde **//servidorA/samba-compartida** es el recurso compartido remoto que está montando y **/mnt/smb** es el punto de montaje.

Para desmontar este directorio ejecute **umount**:

```
[root@clienteB ~]# umount /mnt/smb
```

CREACIÓN DE USUARIOS DE SAMBA

Cuando se configura para ello, Samba respetará las solicitudes hechas por usuarios almacenados en bases de datos de usuarios que a su vez se almacenan en servicios de segundo plano, por ejemplo, LDAP (**Idapsam**), archivos planos (**smbpasswd**) o MySQL (**mysqlsam**).

Aquí agregaremos a la base de datos de Samba un usuario que ya exista en el archivo local **/etc/passwd**. Utilizaremos el formato nativo-predeterminado de la base de datos de usuarios de Samba (el archivo **smbpasswd**) para fines de demostración, pues otras posibilidades disponibles rebasan el alcance de este capítulo.

Generemos una entrada para el usuario yyang. También asignaremos la contraseña del usuario en Samba.

Utilice el comando **smbpasswd** a efecto de crear una entrada para el usuario yyang. Introduzca una contraseña segura cuando se le pregunte. Escriba

```
[root@servidorA ~]# smbpasswd -a yyang
```

New SMB password:

Retype new SMB password:

Added user yyang.

Es posible observar la entrada recién creada para el usuario yyang en el archivo **/etc/samba/smbpasswd**. Escriba

```
[root@servidorA ~]# cat /etc/samba/smbpasswd
```

```
yyang:500:93E28745F5C46E5169E95691975D12BC:[U]
```

] :LCT-42B0250C:

Con un usuario de Samba recién creado, ahora puede hacer que los recursos compartidos sólo estén disponibles para aquellos usuarios autenticados como el recién creado usuario yyang.

Si el usuario yyang ahora quiere acceder a un recurso compartido en el servidor Samba que se ha configurado para su uso exclusivo (un recurso protegido, no público o privado), el usuario ahora puede utilizar el comando **smbclient** como se muestra enseguida:

```
[root@clienteB ~]# smbclient -Uyyang -L //servidorA
```

Desde luego, también es posible acceder a este recurso de Samba compartido y protegido desde una máquina con MS Windows. Pero, cuando se le pregunte, tendrá que proporcionar el nombre de usuario de Samba, además de la contraseña correspondiente, en el cuadro de diálogo para la conexión desde Windows.

Uso de contraseñas NULL

Si necesita permitir que los usuarios no tengan contraseñas (lo cual, por cierto, es una pésima idea), puede hacerlo con el programa **smbpasswd** con la opción **-n**, como sigue:

```
[root@servidorA ~]# smbpasswd -n nombreusuario
```

donde **nombreusuario** es el nombre del usuario cuya contraseña quiere anular.

Por ejemplo, para permitir que yyang entre al recurso compartido en el servidor Samba con una contraseña nula, escriba

```
[root@servidorA ~]# smbpasswd -n yyang  
User yyang password set to none.
```

También puede realizar esta tarea con el programa SWAT a través de su interfaz de Web.

Cambio de contraseñas con smbpasswd

Los usuarios que prefieren trabajar con una línea de comandos en vez de una interfaz de Web pueden utilizar el comando **smbpasswd** para cambiar sus contraseñas de Samba. Este programa trabaja tal como el comando regular **passwd**, excepto que esta variante *no* actualiza el archivo **/etc/passwd** en forma predeterminada. Debido a que **smbpasswd** utiliza el protocolo estándar para comunicarse con el servidor para asuntos relacionados con cambios en contraseñas, también puede utilizarlo con objeto de cambiar la contraseña de una máquina remota que usa Windows.

Por ejemplo, para cambiar la contraseña Samba del usuario yyang, utilice el siguiente comando:

```
[root@servidorA ~]# smbpasswd yyang  
New SMB password:  
Retype new SMB password:
```

Samba puede configurarse para permitir que los usuarios regulares cambien por sí mismos sus contraseñas con el comando **smbpasswd**; el único detalle es que deben conocer su contraseña previa.

USO DE SAMBA PARA AUTENTICACIÓN CONTRA UN SERVIDOR WINDOWS

Hasta ahora hemos hablado de cómo utilizar Samba en un ambiente Samba/Linux. O, para ponerlo en términos reales, hemos estado utilizando Samba en su ambiente nativo, donde es amo y señor de su *dominio*. Lo que esto significa es que nuestro servidor Samba, en combinación con el servidor Linux, ha sido responsable de manejar todos los asuntos de autenticación y autorización de usuarios.

El sencillo esquema de Samba que pusimos en marcha páginas atrás en este capítulo tenía su propia base de datos de usuarios que lograba representar usuarios de Samba como usuarios reales de Linux/UNIX. Esto permitía que cualesquiera archivos y directorios creados por usuarios de Samba tuvieran los contextos de propiedad correctos. Pero, ¿qué pasaría si quisieramos poner en marcha un servidor Samba en un ambiente en el que existieran servidores Windows que se utilizaran para manejar todas las cuentas de usuarios en el dominio? ¿Y si no quisieramos manejar bases de datos de usuarios separadas en Samba? Que aparezca... el demonio **winbindd**.

El demonio **winbindd** se utiliza para traducir información sobre cuentas de usuarios (usuarios y grupos) desde servidores nativos de Windows. También se puede utilizar para traducir otro tipo de información de sistema. Puede hacer esto gracias al uso de (i) **pam_winbind** (un módulo PAM que interactúa con el demonio **winbindd** y que ayuda a autenticar usuarios utilizando la autenticación NT/LM de Windows), (ii) la herramienta **ntlm_auth** (una herramienta que permite el acceso externo a las funciones de autenticación NT/LM de **winbind**), y (iii) el servicio **libnss-winbind** (la biblioteca del Comutador del Servicio de Nombres de **winbind**).

Los pasos para poner en marcha una máquina Linux que consulte a un servidor Windows para la autenticación de sus usuarios son bastante claros y directos. Pueden resumirse de la siguiente manera:

1. Configure las directrices correctas en el archivo de configuración de Samba (**smb.conf**).
2. Agregue **winbind** al servicio del Comutador del Servicio de Nombres del sistema Linux.
3. Una al servidor Samba/Linux con el dominio Windows.
4. Haga pruebas.

Enseguida le presentamos un escenario de muestra en el cual se requiere que un servidor Linux llamado **servidorA** utilice un servidor Windows para asuntos de autenticación de usuarios. El servidor Samba actuará como miembro del dominio del servidor Windows. Suponemos que el servidor Windows que existe para este ejemplo está ejecutando el sistema operativo Windows NT/200x Server, y también es un controlador de dominio (al igual que un servidor WINS). Su dirección IP es 192.168.1.100. El controlador de dominio está operando en modo mixto (la operación en modo mixto proporciona compatibilidad con versiones anteriores de dominios tipo Windows NT y dominios tipo Windows 200x). El nombre del dominio Windows es "WINDOWS-DOMAIN". Hemos convertido en comentario toda definición de recursos compartidos en la configuración de nuestro servidor Samba, así que deberá crear y especificar sus propias definiciones por su cuenta (vea las partes anteriores de este capítulo para obtener detalles sobre la forma de hacerlo). Revisemos el proceso con mayor detalle:

1. Primero cree un archivo **smb.conf** similar a este:

```
#Archivo smb.conf de muestra
[global]
workgroup = WINDOWS-DOMAIN
security = DOMAIN
```

```
username map = /etc/samba/smbusers
log file = /var/log/samba/%m
smb ports = 139 445
name resolve order = wins bcast hosts
wins server = 192.168.1.100
idmap uid = 10000-20000
idmap gid = 10000-20000
template primary group = "Domain Users"
template shell = /bin/bash
winbind separator = +

# Definiciones de recursos compartidos
#[homes]
#  comment = Home Directories
#  browseable = no
#  writable = yes
```

2. Edite el archivo **/etc/nsswitch.conf** que reside en el servidor Linux de manera que tenga entradas similares a estas:

```
passwd: files winbind
shadow: files winbind
group: files winbind
```

3. En un sistema Fedora Core o RHEL, inicie el demonio **winbindd** utilizando el comando **service**. Escriba

```
[root@servidorA ~]# service winbind start
Starting Winbind services: [ OK ]
```

4. Una el servidor Samba al dominio Windows utilizando el comando **net**. Suponga la contraseña de la cuenta del Administrador de Windows y escriba

```
[root@servidorA ~]# net rpc join -U root%contraseña_administrador_windows
Joined domain WINDOWS-DOMAIN
```

donde **contraseña_administrador_windows** es la contraseña de la cuenta en el dominio MS Windows que tiene permiso para unir sistemas al dominio.

5. Utilice la herramienta **wbinfo** para listar todos los usuarios disponibles en el dominio Windows para asegurarse que todo está funcionando como debe ser. Escriba

```
[root@servidorA ~]# wbinfo -u
```

LOCALIZACIÓN DE FALLAS EN SAMBA

Existen cuatro soluciones típicas para problemas de conectividad con Samba.

- ▼ **Reinic peace Samba** Quizá sea necesario hacerlo si es que Samba entró en un estado de indefinición o (aun más probable) usted hizo cambios a la configuración pero olvidó reiniciar Samba para que dichos cambios tuvieran efecto.

- **Asegúrese de que las opciones de configuración son correctas** Es típico encontrar errores en el archivo de configuración **smb.conf**, sobre todo en nombres de directorios, nombres de usuarios, números de red y nombres de anfitriones. Un descuido frecuente es cuando añade un nuevo cliente a un grupo que cuenta con acceso especial al servidor pero no le avisa a Samba el nombre del nuevo cliente que agrega. Recuerde que, para errores de sintaxis, la herramienta **testparm** es su aliada.
- ▲ **Monitoree contraseñas con encriptación** Quizá no coincidan, cuando configura al servidor para utilizarlas pero no a los clientes o (lo más seguro) los clientes están utilizando contraseñas con encriptación y Samba no se ha configurado para ello. Si está bajo presión para poner en marcha un cliente, quizás sólo quiera inhabilitar la encriptación del lado del cliente utilizando los scripts **regedit** que vienen con el código fuente de Samba (vea el subdirectorio **docs**).

RESUMEN

En este capítulo abordamos el proceso de compilación, instalación y configuración de Samba, de manera que su servidor Linux pueda integrarse a una red basada en Windows. Samba es una poderosa herramienta que tiene potencial suficiente como para reemplazar servidores MS Windows que están dedicados a compartir recursos de disco e impresión.

Leer toneladas de documentación quizás no sea su manera favorita de pasar el día, pero hallará que la documentación de Samba es completa, provechosa y de fácil lectura. Cuando menos escudriñe por esos archivos para ver qué contienen, de manera que sepa dónde puede buscar en caso de requerir información adicional de primera. Con una gran variedad de textos sobre Samba (gratis y de paga) que hoy están disponibles, cuenta con todo lo necesario para configurar aun los más complejos escenarios. De inmediato se tienen presentes dos excelentes obras dedicadas a Samba: *Samba-3 by Example*, de John Terpstra; y *The Official Samba-3 HOWTO and Reference Guide*, de John Terpstra y Jelmer Vernooij, ambas publicadas por Prentice-Hall (marzo de 2004). Están disponibles en versión impresa y electrónica. Las versiones electrónicas de las obras están disponibles en <http://www.samba.org>.

CAPÍTULO 25



LDAP

Se le ha llamado de distintas formas al Lightweight Directory Access Protocol (LDAP) (Protocolo ligero de acceso a directorios), incluso como el mejor invento desde la aparición del pan rebanado para emparedados. En realidad se trata de un conjunto de protocolos abiertos utilizados para acceder y modificar información almacenada y centralizada en una red. LDAP se basa en el estándar X.500 (X.500 es un estándar ISO que define un modelo universal para servicios de directorios distribuidos) pero es una versión más ligera que el estándar original. La RFC 2251 explica la relación así: "LDAP está diseñado para proporcionar acceso a directorios que respaldan modelos X.500, pero sin incurrir en los requerimientos de recursos que impone el protocolo de acceso a directorios X.500. Al igual que las bases de datos tradicionales, una base de datos LDAP se puede consultar para extraer la información que almacena".

LDAP fue creado por la Universidad de Michigan en 1992 como una alternativa ligera al Directory Access Protocol (DAP) (Protocolo de acceso a directorios). Por sí mismo, LDAP no define el servicio de directorios. En vez de ello, define el transporte y el formato de mensajes utilizado por un cliente para acceder a los datos en un directorio (un directorio X.500, por ejemplo).

LDAP es ampliable, relativamente fácil de implementar y se basa en un estándar abierto, es decir, no propietario (que no es propiedad de alguien en particular). Este capítulo le presentará una introducción al mundo de los servicios de directorios, según la implementación OpenLDAP. Por tal motivo, abordaremos los conceptos esenciales que gobiernan la arquitectura y uso de LDAP.

FUNDAMENTOS DE LDAP

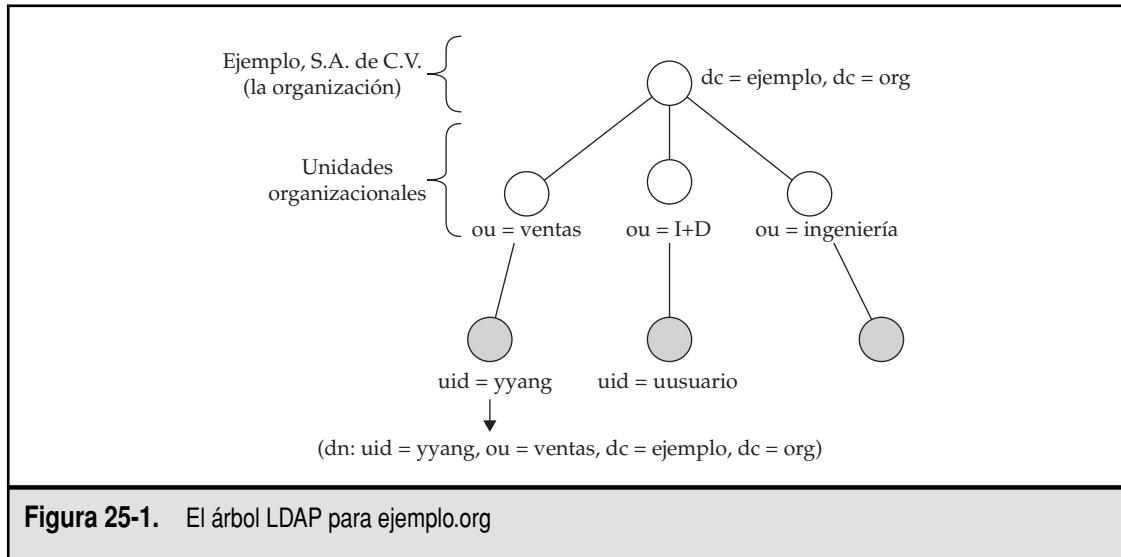
LDAP es un servicio global de directorios. Este directorio se puede usar para almacenar todo tipo de información. Se le puede considerar como una base de datos de objetos de distintas clases. Pero, a diferencia de las bases de datos tradicionales, una base de datos LDAP es especialmente apta para operaciones de lectura, búsqueda y navegación en vez de serlo para operaciones de escritura. Son las lecturas las que le dan a LDAP su más reluciente fama.

Enseguida mostramos algunas implementaciones populares de este protocolo:

- ▼ OpenLDAP es, como su nombre lo indica, un conjunto abierto de LDAP.
- Netware Directory Service de Novell, también conocido como eDirectory.
- Active Directory de Microsoft.
- iPlanet Directory Server, el cual se dividió años atrás entre Sun y Netscape. Desde entonces, la tecnología Netscape Directory Server la adquirió Red Hat, que a su vez la entregó a la comunidad Open Source.
- ▲ SecureWay Directory de IBM.

El directorio LDAP

Al igual que en el popular Domain Name Service (DNS) (Servicio de nombres de dominio, visto en el capítulo 16), las entradas de un directorio LDAP están estructuradas en forma de árbol jerárquico. Como sucede en muchas estructuras jerárquicas, tanto mayor sea la profundidad del árbol, mayor será la precisión del contenido almacenado en ella. A la estructura de árbol jerárquico de LDAP se le conoce de manera formal como *directory information tree (DIT)* (*árbol de información del directorio*)



directorio). A la parte superior de esta estructura jerárquica se le conoce como el elemento *raíz*. La ruta completa hacia cualquier nodo en la estructura del árbol, misma que define a ese nodo en forma única, se le conoce como *distinguished name (DN) (nombre diferenciado)* del nodo u objeto.

Al igual que en DNS, es usual que la estructura de un directorio LDAP refleje límites geográficos u organizacionales. Los límites geográficos pueden ocurrir en las líneas divisorias de un país, de un estado, de una ciudad, y demás. Los límites organizacionales pueden, por ejemplo, referirse a líneas divisorias entre funciones, departamentos o unidades de organización.

Por ejemplo, una empresa llamada Ejemplo, S.A. de C.V., podría estructurar su directorio utilizando una estructura de nombres basada en dominios. Esta compañía tiene diferentes subdivisiones o unidades organizacionales (OU, organizational units) dentro de sí, como el departamento de Ingeniería, el departamento de Ventas y el departamento de Investigación y Desarrollo. El directorio LDAP para una compañía como tal se muestra en la figura 25-1.

El DN para el objeto que se tomó como muestra en el directorio mostrado en esa figura es “dn:uid=yyang,ou=ventas,dc=ejemplo,dc=org”.

Modelo cliente/servidor

Como sucede con la mayoría de los servicios de red, LDAP se apega al paradigma cliente/servidor. En este contexto, una interacción típica entre el cliente y el servidor sería como se describe a continuación:

- ▼ La aplicación de un cliente de LDAP se conecta al servidor LDAP. A ello se le conoce a veces como atarse o enlazarse al servidor.
- Dependiendo de las restricciones de acceso configuradas en el servidor LDAP, éste tiene dos opciones: acepta la solicitud de enlace/conexión, o la rechaza. Suponiendo que acepta...

- El cliente entonces tiene las opciones de consultar al servidor del directorio, escudriñar la información almacenada en el servidor, o intentar la modificación/actualización de la información en el servidor LDAP.
- ▲ Si, por el contrario, basándose en las restricciones de acceso definidas, el servidor rechaza cualquiera de las operaciones definidas por el cliente, puede redirigir o referir a dicho cliente hacia un servidor LDAP superior que quizás tenga mayor autoridad para atender la solicitud.

Usos de LDAP

LDAP es un servicio de directorios distribuidos y como tal puede utilizarse para almacenar varios tipos de información. Prácticamente se puede almacenar cualquier clase de información en un directorio LDAP, tan variada en su naturaleza como texto, imágenes, datos binarios o certificados públicos.

Al pasar de los años se han creado varios esquemas LDAP para almacenar diferentes gamas de datos en un directorio LDAP. Enseguida listamos algunos ejemplos de los usos de LDAP:

- ▼ LDAP puede servir como una completa Solución para la Administración de Identidad dentro de una organización. Puede proporcionar servicios de autenticación y autorización para usuarios. De hecho, los servicios proporcionados por el Servicio de información en red (NIS) visto en el capítulo 23, se pueden reemplazar por completo con LDAP.
- La información almacenada en registros DNS se puede almacenar en LDAP.
- LDAP se puede usar para proporcionar servicios de “páginas amarillas” dentro de una organización (por ejemplo, para proveer la información de contacto de los usuarios o empleados, números telefónicos, direcciones, departamentos, etc.).
- La información de ruta para correo electrónico se puede almacenar en LDAP.
- ▲ Existe un esquema en Samba que permite que un servidor Samba almacene grandes cantidades de objetos y sus respectivos atributos en LDAP. Esto permite que Samba pueda operar como un robusto reemplazo de los controladores de dominio de Windows NT en ambientes donde se necesitan la redundancia y la duplicación.

Terminología de LDAP

Si habrá de graduarse en el dialecto del LDAP, cuando menos deberá conocer la jerga técnica esencial de este protocolo. En esta sección intentaremos definir algunos términos que con frecuencia encontrará al lidiar con LDAP.*

- ▼ **Entrada u objeto** La unidad mínima en un directorio LDAP es un objeto. A cada objeto se le llama por su nombre diferenciado (DN). Por ejemplo, el objeto que se utilizó de muestra en la figura 25-1 se llama “dn:uid=yyang,ou=ventas,dc=ejemplo,dc=org”.
- **Atributos** Estas son las piezas de información asociadas a un objeto. Por ejemplo, la dirección de una organización o el puesto y los números telefónicos de un empleado.

* La terminología LDAP guarda una estrecha relación con los conceptos utilizados por el paradigma de Diseño Orientado a Objetos, todavía vigente entre los practicantes de la ingeniería de software; ello se debe a que LDAP se basa en X.500, que es un estándar orientado a objetos por naturaleza (N. del T.).

- **objectClass** Éste es un tipo especial de atributo. Todos los objetos en LDAP deben tener el atributo objectClass. La definición de objectClass especifica cuáles atributos requiere un objeto LDAP, así como las clases de objetos que pueden existir. Los valores de este atributo los pueden modificar los clientes, pero el atributo objectClass en sí no puede eliminarse.

Las definiciones de objectClass se almacenan por sí mismas en archivos de esquemas.

- **Esquema (Schema)** Un esquema es una colección de reglas que determinan la estructura y contenido de un directorio. El esquema contiene las definiciones, los tipos de atributo y de las clases de objetos, etcétera.

Además de listar los atributos para cada clase de objeto, un esquema define si esos atributos son opcionales u obligatorios. Es usual que los esquemas se almacenen en archivos de texto plano.

Ejemplos de esquemas son:

- ▼ **core.schema** Este esquema define los objetos y atributos básicos de LDAPv3. Es un esquema central requerido en la implementación de OpenLDAP.
- ▲ **inetorgperson.schema** Define la clase de objetos inetOrgPerson y sus atributos asociados. Con frecuencia este objeto se utiliza para almacenar información de contacto de personas.
- ▲ **LDIF** Siglas de LDAP Data Interchange Format (Formato de intercambio de datos LDAP). Es un archivo de texto plano para objetos LDAP. Los archivos que importan o exportan datos hacia y desde un servidor LDAP deben hacerlo con este formato. Los datos utilizados para la duplicación entre servidores LDAP también están en este formato.

OPENLDAP

OpenLDAP es una implementación de fuente abierta de LDAP que funciona en sistemas Linux/UNIX. Es un conjunto de programas integrado por los siguientes componentes: **slapd**, **slurpd** y **libraries**, los cuales implementan el protocolo LDAP, y varias herramientas más para uso tanto del lado del cliente como del lado del servidor.

Demonios utilizados del lado del servidor

El lado del servidor consta de dos demonios principales:

- ▼ **Slapd** Éste es un demonio LDAP independiente. Escucha las conexiones LDAP hechas desde los clientes y atiende las operaciones que recibe sobre dichas conexiones.
- ▲ **Slurpd** Éste es un demonio LDAP independiente para la duplicación. Se usa para propagar cambios de una base de datos **slapd** a otra. Es un demonio que se usa para sincronizar cambios de un servidor LDAP a otro. Sólo se requiere cuando están en uso dos o más servidores LDAP.

Utilidades OpenLDAP

Las utilidades OpenLDAP son un conjunto de herramientas que funcionan desde una línea de comandos y se usan para consultar, visualizar, actualizar y modificar datos almacenados en un directorio OpenLDAP. En sistemas Fedora Core y RHEL, este conjunto de programas se envía en el paquete **openldap-clients*.rpm**, y algunas de ellas se envían en el paquete **openldap-server*.rpm**. Los programas se listan en la tabla 25-1.

Instalación de OpenLDAP

A fin de poner en marcha los componentes OpenLDAP para el cliente y el servidor, los siguientes paquetes se requieren en sistemas Fedora y Red Hat Enterprise Linux:

- ▼ **openldap-2*.rpm** Proporciona archivos de configuración y las bibliotecas para OpenLDAP.
- **openldap-clients*.rpm** Proporciona programas que funcionan al lado del cliente, necesarios para acceder y modificar directorios OpenLDAP.
- ▲ **openldap-servers*.rpm** Proporciona programas que corren del lado del servidor (**slapd**, **slurpd**) y otras utilidades necesarias para configurar y ejecutar LDAP.

SUGERENCIA Si sólo está configurando el lado del cliente, no necesitará el paquete **openldap-servers*.rpm**.

Utilizaremos el programa **up2date** para descargar e instalar en forma automática el paquete **openldap-servers** en nuestro sistema muestra. Enseguida listamos los pasos:

1. Una vez dentro de una sesión de trabajo del usuario raíz, confirme cuáles de los paquetes ya tiene instalado el sistema consultando la base de datos RPM.

```
[root@servidorA ~]# rpm -qa | grep -i openldap  
openldap-2*  
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
```

NOTA El proceso de instalación de la mayoría de las distribuciones de Linux incluirá en forma automática la base del programa OpenLDAP como parte del software mínimo a instalar. Esto se hace para que el sistema se pueda configurar como un cliente de LDAP desde que concluye la instalación, sin ninguna molestia adicional.

2. Nuestro sistema de muestra ya tiene instaladas las bibliotecas **openldap** básicas, así que procederemos a la instalación de los paquetes OpenLDAP del cliente y del servidor utilizando **up2date**. Escriba

```
[root@servidorA ~]# up2date -i openldap-servers openldap-clients
```

3. Una vez concluida en forma exitosa la instalación, puede pasar a la sección de configuración.

Utilidad	Descripción
ldapmodify	Utilizada para modificar objetos en LDAP. Acepta entradas directamente desde una línea de comandos o mediante un archivo.
ldapadd	Este comando en realidad es un vínculo duro al comando ldapmodify -a . Se usa para añadir nuevos objetos a una base de datos LDAP (la funcionalidad proporcionada por el comando ldapadd se puede obtener al añadir la opción -a al comando ldapmodify).
ldapdelete	Utilizado para eliminar objetos de un directorio de LDAP.
ldappasswd	Establece la contraseña de un usuario LDAP.
ldapsearch	Utilizado para consultar/buscar un directorio LDAP.
slapadd	Acepta entradas desde un archivo LDIF para poblar un directorio LDAP. Ubicado dentro del directorio /usr/sbin .
slapcat	Vacia todo el contenido del directorio LDAP en un archivo tipo LDIF. Ubicado dentro del directorio /usr/sbin .
slapindex	Utilizado para volver a indexar la base de datos LDAP con el contenido actual de la base de datos. Ubicado dentro del directorio /usr/sbin .
slappasswd	Utilizado para generar contraseñas con encriptación apropiada y mezclada que se puedan usar en varias operaciones privilegiadas con directorios. Ubicado dentro del directorio /usr/sbin .

Tabla 25-1. Utilidades OpenLDAP

Configuración de OpenLDAP

Dependiendo de lo que quiera hacer con su directorio, la configuración del servidor donde éste residirá puede convertirse en un verdadero dolor o ser un proceso muy simple. Lo más usual es que poner en marcha su servidor será muy fácil si está trabajando en una instalación nueva en la que no tiene que preocuparse de asuntos sobre compatibilidad con arquitecturas legadas, usuarios o datos existentes, etc. Para ambientes con infraestructura existente, deben tomarse medidas precautorias adicionales.

CUIDADO Si está instalando LDAP en un ambiente en el que sí tiene que preocuparse de asuntos sobre compatibilidad con versiones anteriores, arquitecturas legadas, usuarios o datos existentes, entonces se le aconseja que proceda con el despliegue de OpenLDAP con extremada cautela. En algunas situaciones ello puede tomar meses de planeación, pues es preciso considerar exhaustivas pruebas en paralelo en sistemas alternos antes de liberar el funcionamiento del ambiente LDAP que quiere implementar.

Los módulos pam_ldap y nss_ldap

El módulo **pam_ldap** proporciona los medios para que los anfitriones Linux/UNIX puedan autenticar usuarios contra directorios LDAP. Fue creado por la compañía de software PADL (<http://www.padl.com>). Permite que las aplicaciones habilitadas para el uso de PAM puedan autenticar usuarios utilizando información almacenada en un directorio LDAP. Ejemplos de dichas aplicaciones son los programas de registro, algunos servidores de correo electrónico, algunos servidores FTP, OpenSSH y Samba.

El módulo **nss_ldap** es una biblioteca de extensiones en C que permiten a las aplicaciones buscar usuarios, grupos, anfitriones, y otra información consultando un directorio LDAP. El módulo permite a las aplicaciones realizar búsquedas de información utilizando LDAP al igual que mediante los métodos tradicionales como lo son archivos planos o NIS. El módulo también lo creó la compañía de software PADL (<http://www.padl.com>).

En sistemas Fedora y RHEL, el paquete **nss_ldap*.rpm** proporciona estos módulos, los cuales se necesitan en sistemas donde LDAP se utiliza para reemplazar los mecanismos tradicionales de autenticación.

Veamos si el paquete ya está instalado escribiendo la siguiente instrucción

```
[root@servidorA openldap]# rpm -q nss_ldap  
nss_ldap-*
```

Si se da cuenta de que el paquete aún no está instalado, lo puede hacer rápidamente con el uso del comando **up2date**, como sigue:

```
[root@servidorA openldap]# up2date -i nss_ldap
```

Otro factor de suma importancia al que debe prestar atención cuando configura su servicio de directorio LDAP es la estructura del directorio en sí. Por ejemplo, debe tener respuesta a las siguientes preguntas antes de proceder: “¿Cuáles son las divisiones organizacionales en su particular caso?” “¿Sobre cuáles líneas divisorias se construirá la estructura?” Otras preguntas que también debe tener presentes son: “¿Qué tan delicada es la información que quiere almacenar en el directorio?” “¿Será necesario utilizar más de un servidor LDAP?”

Configuración de slapd

El archivo **/etc/openldap/slapd.conf** es el archivo de configuración para el demonio **slapd**. En esta sección haremos el análisis munucioso del archivo de configuración predeterminado que viene con nuestro sistema Fedora y abordaremos algunas de sus porciones más interesantes.

Enseguida podrá ver una versión trunca del archivo **/etc/openldap/slapd.conf**. La mayoría de las entradas comentadas en el archivo original se han eliminado, al igual que otras directrices de configuración a las que no habremos de referirnos ahora. Mostramos la versión recortada del archivo que es relevante para nuestra discusión actual. También hemos añadido la numeración de las líneas a fin de facilitar su legibilidad.

```
1 # Vea slapd.conf(5) para obtener más detalles sobre las opciones de configuración.
2 # Este archivo NO se debe poner a disposición del público.
3 #
4 include      /etc/openldap/schema/core.schema
5 include      /etc/openldap/schema/cosine.schema
6 include      /etc/openldap/schema/inetorgperson.schema
7 include      /etc/openldap/schema/nis.schema
8 #
9 pidfile      /var/run/slapd.pid
10 argsfile     /var/run/slapd.args
11 database     bdb
12 suffix       "dc=my-domain,dc=com"
13 rootdn       "cn=Manager,dc=my-domain,dc=com"
14 # Contraseñas en texto plano, sobre todo para la directriz rootdn, deben
15 # evitarse. Vea slappasswd(8) y slapd.conf(5) para más detalles.
16 #
17 rootpw        {crypt}ijFYNCsNctBYg
18 #
19 # El directorio de la base de datos DEBE existir previa ejecución de slapd y
20 # sólo debe permitir el acceso a slapd y a las herramientas slap.
21 # Se recomienda utilizar el modo 700.
22 directory     /var/lib/ldap
```

Del listado anterior:

- ▼ Las líneas 1-3 son comentarios. Cualquier texto después del signo de número (#) es un comentario.
- Las líneas 4-7 contienen instrucciones **include**. La instrucción **include** indica a **slapd** que lea la información sobre configuraciones adicionales de los archivos especificados. En este caso se trata de archivos que contienen esquemas OpenLDAP almacenados dentro del directorio **/etc/openldap/schema/**. Cuando menos el archivo **core.schema** debe estar presente.
- En la línea 9, la directriz **pidfile** apunta hacia la ruta del archivo que contendrá el identificador del proceso de **slapd**.
- En la línea 10, la directriz **argsfile** se usa para especificar la ruta hacia el archivo que puede utilizarse para almacenar opciones de arranque de **slapd** cuando se inicia desde la línea de comandos.
- En la línea 11, la opción **database** marca el inicio de una nueva definición de la instancia de la base de datos. El valor de esta opción depende del programa de segundo plano que se utilice para contener a la base de datos. En nuestro archivo **slapd.conf** ejemplo, **bdb** (Berkeley DB) se usa como el tipo de base de datos. Otros programas de bases de datos de segundo plano para los que se ofrece soporte son **ldbm**, **sql**, **tcl** y **meta**. En la tabla 25-2 se describen algunas bases de datos de segundo plano.
- ▲ En la línea 12, la directriz **suffix** especifica el sufijo DN de las consultas que se pasarán a la base de datos de segundo plano elegida. Define el dominio para el cual el servidor LDAP proporciona información o para el cual tiene la autoridad el servidor LDAP. Esta entrada se debe cambiar para reflejar la estructura de nombres de su organización.

Tipo de base de datos de segundo plano	Descripción
bdb	Definición de instancia para la base de datos Berkeley. Este es el tipo de base de datos de segundo plano que se recomienda. Utiliza el Sleepycat Berkeley DB para almacenar datos.
ldbm	Este tipo significa LDAP DBM (DBM del Protocolo ligero de acceso a directorios). Es muy fácil de configurar pero no es tan durable como el tipo bdb previamente descrito. También utiliza Berkeley DB, GNU DBM y MDBM para almacenar datos.
sql	Utiliza una base de datos SQL de segundo plano para almacenar datos.
ldap	Utiliza un proxy para redirigir solicitudes entrantes hacia otro servidor LDAP.
meta	Base de datos de segundo plano de metadirectorio. Es una mejora al tipo ldap de segundo plano. Lleva a cabo la redirección hacia un proxy LDAP respecto de un conjunto de servidores LDAP remotos.
monitor	Almacena información acerca del estado del demonio slapd .
null	Operaciones solicitadas a esta base de datos concluyen con éxito pero no hacen nada. Ello es equivalente a enviar cualquier cosa a /dev/null en Linux/UNIX.
passwd	Utiliza el archivo de texto plano /etc/passwd del sistema para proveer información sobre cuentas de usuario.
tcl	Una base de datos experimental de segundo plano que utiliza un intérprete Tcl que está incrustado en forma directa en slapd .
perl	Utiliza un intérprete Perl que está incrustado en forma directa en slapd .

Tabla 25-2. Bases de datos de segundo plano soportadas por OpenLDAP y otras opciones

- En la línea 13, la directriz **rootdn** especifica el DN del superusuario para el directorio LDAP. Este usuario es para el directorio LDAP lo que el usuario raíz de Linux/UNIX es para un sistema Linux. El usuario especificado aquí no está sujeto a controles de acceso o a restricciones administrativas para realizar operaciones en la base de datos en cuestión. El DN especificado no necesariamente tiene que existir en el directorio.
- En la línea 17, la directriz **rootpw** especifica la contraseña para el DN determinado por la directriz **rootdn**. No hace falta decirlo, aquí debe utilizar una contraseña profesional

muy fuerte. La contraseña puede especificarse en texto plano (pésima, nefasta y terrible idea); o bien, el resultado de mezclar la contraseña se puede especificar aquí. El programa **slappasswd** se puede usar para generar mezclas de contraseñas.

- ▲ Por último, en la línea 22, la directriz **directory** especifica la ruta hacia los archivos BDB que contienen la base de datos y sus índices asociados.

Una vez explicadas estas importantes directrices del archivo **slapd.conf**, haremos unos cuantos cambios al archivo para personalizarlo a nuestro ambiente.

- ▼ Mientras está registrado en el sistema como el usuario raíz, entre al directorio de trabajo de OpenLDAP. Escriba

```
[root@servidorA ~]# cd /etc/openldap/
```

- Renombre el archivo **slapd.conf** original para hacer un respaldo. Esto es para que siempre pueda revertir los cambios en caso de que cometa errores. Escriba

```
[root@servidorA openldap]# mv slapd.conf slapd.conf.original
```

- Utilice cualquier editor de textos para crear el archivo **/etc/openldap/slapd.conf** utilizando el siguiente texto:

```
include      /etc/openldap/schema/core.schema
include      /etc/openldap/schema/cosine.schema
include      /etc/openldap/schema/inetorgperson.schema
include      /etc/openldap/schema/nis.schema
pidfile     /var/run/slapd.pid
argsfile    /var/run/slapd.args
database   bdb
suffix      "dc=ejemplo,dc=org"
rootdn     "cn=MeroMero,dc=ejemplo,dc=org"
#
# La contraseña mezclada mostrada enseguida se generó con el siguiente comando:
# "slappasswd -s test". Ejecute el comando y pegue la salida aquí.
rootpw  {SSHA}gJeD9BJdcx5L+bfgMpmvsFJVqdG5CjdP
directory  /var/lib/ldap
```

- ▲ Guarde los cambios en el archivo y salga del editor.

Inicio y terminación de **slapd**

Después de preparar el archivo de configuración de **slapd**, el siguiente paso será que inicie el demonio. En un sistema Fedora, empezarlo es muy sencillo. Pero antes utilizaremos el comando **service** para verificar el estado del demonio.

```
[root@servidorA ~]# service ldap status
slapd is stopped
```

La salida de nuestro ejemplo muestra que el demonio no está en operación en este momento. Inícielo con el siguiente comando:

```
[root@servidorA openldap]# service ldap start
Checking configuration files for slapd: config file testing succeeded
[    OK    ]
Starting slapd:                                         [    OK    ]
```

Y si encuentra que el servicio **ldap** ya está en operación, entonces puede emitir el comando **service** con la opción **restart**, como sigue:

```
[root@servidorA ~]# service ldap restart
Stopping slapd:                                         [    OK    ]
Checking configuration files slapd: config file testing succeeded [ OK ]
Starting slapd:                                         [    OK    ]
```

SUGERENCIA Vigile los permisos del archivo de configuraciones de OpenLDAP. Por ejemplo, el demonio **slapd** rehusará iniciar en un sistema Fedora o RHEL si el usuario “**ldap**” no puede leer el archivo **slapd.conf**. Asimismo, el contenido de la base de datos del directorio (**/var/lib/ldap**) debe ser propiedad de “**ldap**” a fin de evitar errores extraños.

Si quiere que **slapd** inicie en forma automática la próxima vez que reinicie el sistema, escriba

```
[root@servidorA openldap]# chkconfig ldap on
```

CONFIGURACIÓN DE CLIENTES DE OPENLDAP

Toma algún tiempo adquirir la noción de clientes en el mundo de LDAP. Casi cualquier recurso del sistema o proceso puede ser un cliente de LDAP. Y, afortunadamente o no, cada grupo de clientes tiene sus propios archivos específicos de configuración. Es común que este tipo archivos para clientes de OpenLDAP se llame **ldap.conf**. Pero se almacenan en distintos directorios dependiendo del cliente en particular en cuestión.

Dos ubicaciones comunes para los archivos de configuración de clientes de OpenLDAP son los directorios **/etc/openldap/** y **/etc/**. Las aplicaciones de clientes que utilizan bibliotecas OpenLDAP (proporcionadas por el paquete **openldap*.rpm**), programas tales como **ldapadd**, **ldapsearch**, **Sendmail** y **Evolution** consultan el archivo **/etc/openldap/ldap.conf**, si es que existe. Las bibliotecas **nss_ldap**, a su vez, utilizan el archivo **/etc/ldap.conf** como su archivo de configuración.

En esta sección haremos los ajustes necesarios para los archivos de configuración de las herramientas de cliente de OpenLDAP. Este archivo de configuración es bastante claro y directo; nada más cambiaremos una de las directrices en el archivo.

Abra el archivo **/etc/openldap/ldap.conf** en cualquier editor de texto y cambie esta línea en el listado:

```
BASE dc=example,dc=com
```

a que se vea como sigue:

```
BASE dc=ejemplo,dc=org
```

SUGERENCIA Si está utilizando las herramientas cliente de un anfitrión que no sea el propio servidor, otra variable/directriz particular que deberá cambiar en el archivo `/etc/openldap/ldap.conf` es la directriz HOST. Debería establecerla a la dirección IP del servidor LDAP remoto. Pero, como en este ejemplo estamos utilizando los clientes de LDAP en el propio servidor LDAP, dejaremos la directriz HOST con su valor predeterminado, es decir, `HOST 127.0.0.1`.

Creación de objetos en el directorio

El Data Interchange Format LDAP (LDIF) (Formato de intercambio de datos LDAP) se usa para representar objetos en un directorio LDAP a partir de texto plano. Como se dijo antes, los datos en LDAP se presentan e intercambian en este formato. Los datos contenidos en un archivo LDIF se pueden usar para manipular, agregar, eliminar y cambiar la información almacenada en el directorio LDAP. El formato de una entrada LDIF es el siguiente:

```
dn: <nombre diferenciado>
<atributo_descripción>: <atributo_valor>
<atributo_descripción>: <atributo_valor>

dn: <y otro nombre diferenciado>
<atributo_descripción>: <atributo_valor>
<atributo_descripción>: <atributo_valor>
...
...
```

El archivo LDIF es ligeramente estricto con este formato. Debe tener presentes los siguientes puntos:

- ▼ Varias entradas dentro del mismo archivo LDIF las separan por líneas vacías una de la otra.
- Las entradas que empiezan con un signo de número (#) se consideran como comentarios y se ignoran.
- Una entrada que se extiende más de una línea puede continuar en la siguiente línea si empieza con un espacio en blanco o un carácter de tabulación.
- ▲ El espacio después de los dos puntos (:) es importante para cada entrada.

En esta sección utilizaremos un archivo LDIF de muestra para poblar nuestro nuevo directorio con información básica que servirá para establecer nuestro árbol de información del directorio (DIT, directory information tree) y también para describir un par de usuarios llamados *chon* y *chano*.

1. Enseguida mostramos el archivo LDIF. Utilice cualquier editor de texto para introducir el siguiente texto en el archivo. Sea muy cuidadoso con los espacios en blanco y las tabulaciones en el archivo, y asegúrese de mantener una línea en blanco después de cada entrada DN, como se muestra a continuación en nuestro archivo ejemplo.

```
dn: dc=ejemplo,dc=org
objectclass: dcObject
```

```

objectclass: organization
o: Ejemplo SA de CV
dc: ejemplo

dn: cn=chon,dc=ejemplo,dc=org
objectclass: organizationalRole
cn: chon

dn: cn=chano,dc=ejemplo,dc=org
objectclass: organizationalRole
cn: chano

```

2. Acto seguido, guarde el archivo como **muestra.ldif** y salga del editor de texto.
3. Utilice la utilidad **ldapadd** para importar el archivo **muestra.ldif** en el directorio OpenLDAP. Escriba

```
[root@servidorA ~]# ldapadd -x -D "cn=manager,dc=ejemplo,dc=org" -W -f muestra.ldif
Enter LDAP Password:
adding new entry "dc=ejemplo,dc=org"
adding new entry "cn=bogus,dc=ejemplo,dc=org"
adding new entry "cn=testuser,dc=ejemplo,dc=org"
```

Estos son los parámetros utilizados en este comando **ldapadd**:

- ▼ **x** Significa que debe utilizarse autenticación simple en vez de SASL.
 - **D** Especifica el nombre diferenciado con el cual ligará los datos en el directorio LDAP (es decir, el parámetro especificado en el archivo **slapd.conf**).
 - **W** Permite que se le pregunte al usuario la contraseña de autenticación simple, ello en vez de que dicha contraseña sea especificada en texto plano en la línea de comando.
 - ▲ **f** Especifica el archivo desde el cual leer el archivo LDIF.
4. Introduzca la contraseña que antes creó utilizando la utilidad **slappasswd**, es decir, la contraseña que se especificó en la directriz **rootpw** en el archivo **/etc/openldap/slapd.conf**. Utilizamos “test” como la contraseña para nuestro ejemplo.

Con esto terminamos de poblar el directorio.

Búsqueda, consulta y modificación del directorio

Utilizaremos un par de utilidades para cliente de OpenLDAP que nos servirán para extraer información de nuestro directorio.

1. Primero utilizaremos la utilidad **ldapsearch** para buscar y recuperar todas las entradas (objetos) en el directorio de la base de datos. Escriba

```
[root@servidorA ~]# ldapsearch -x -b 'dc=ejemplo,dc=org' '(objectclass=*)'
# extended LDIF
...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
```

```
# ejemplo.org
dn: dc=ejemplo,dc=org
objectClass: dcObject
objectClass: organization
o: Ejemplo SA de CV
dc: ejemplo

# bogus, ejemplo.org
dn: cn=bogus,dc=ejemplo,dc=org
objectClass: organizationalRole
cn: chon

...<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>...
```

```
# numResponses: 4
# numEntries: 3
```

2. Volvamos a realizar la búsqueda, pero esta vez sin especificar la opción **-b** y también reduciendo la cantidad de texto de salida. Escriba

```
[root@servidorA ~]# ldapsearch -x -LLL '(objectclass=*)'
dn: dc=ejemplo,dc=org
objectClass: dcObject
objectClass: organization
o: Ejemplo SA de CV
dc: ejemplo

dn: cn=chon,dc=ejemplo,dc=org
objectClass: organizationalRole
cn: chon

dn: cn=chano,dc=ejemplo,dc=org
objectClass: organizationalRole
cn: chano
```

No tuvimos que especificar el **basedn** que había que buscar porque esa información ya está definida en nuestro archivo **/etc/openldap/ldap.conf**.

3. Ahora limitaremos nuestra consulta buscando sólo el objeto cuyo nombre común (cn) es igual a *chon*. Para ello, escriba el siguiente comando:

```
[root@servidorA ~]# ldapsearch -x -LLL -b 'dc=ejemplo,dc=org' '(cn=chon)'
dn: cn=chon,dc=ejemplo,dc=org
objectClass: organizationalRole
cn: chon
```

4. Ahora intentaremos llevar a cabo una operación privilegiada sobre un objeto del directorio utilizando la utilidad **ldapdelete**. Eliminemos el objeto cuyo DN es “*cn=chon,dc=ejemplo,dc=org*”. Escriba el siguiente comando:

```
[root@servidorA ~]# ldapdelete -x -W -D 'cn=Manager,dc=ejemplo,dc=org' \
'cn=chon,dc=ejemplo,dc=org'
Enter LDAP Password:
```

Introduzca la contraseña para el DN cn=Manager,dc=ejemplo,dc=org a fin de completar la operación.

- Utilicemos de nuevo la utilidad **ldapsearch** para asegurarnos de que el objeto se eliminó. Escriba

```
[root@servidorA ~]# ldapsearch -x -LLL -b 'dc=ejemplo,dc=org' '(cn=chon)'
```

El comando no debería regresar nada.

Uso de OpenLDAP para autenticación de usuarios

Ahora abordaremos la puesta en marcha de un servidor OpenLDAP (y el cliente) que configuraremos páginas antes en este mismo capítulo para que, además, maneje usuarios de Linux. Utilizaremos algunos de los scripts de migración que vienen con el software para extraer/migrar los usuarios que ya existen en el archivo **/etc/passwd** del sistema en LDAP.

Configuración del servidor

Es muy fácil configurar un sistema Linux como un almacén de segundo plano para información sobre cuentas de usuario, sobre todo si ya tiene resueltos todos los demás asuntos básicos de la configuración de OpenLDAP.

El software viene con varios scripts útiles que facilitan la migración de varias bases de datos en un directorio OpenLDAP. Estos scripts se almacenan dentro del directorio **/usr/share/openldap/migration/** en sistemas Fedora Core o RHEL.

Empecemos por personalizar el archivo **/usr/share/openldap/migration/migrate_common.php** para ajustarlo a nuestras necesidades.

- Abra el archivo para editararlo y busque las líneas/entradas similares a estas:

```
$DEFAULT_MAIL_DOMAIN = "padl.com";
$DEFAULT_BASE = "dc=padl,dc=com";
```

Por ejemplo, cambiaremos estas variables para que lean

```
$DEFAULT_MAIL_DOMAIN = "ejemplo.org";
$DEFAULT_BASE = "dc=ejemplo,dc=org";
```

- Utilicemos uno de los scripts de migración (**migrate_base.pl**) para crear la estructura base de nuestro directorio. Escriba

```
[root@servidorA ~]# cd /usr/share/openldap/migration/
```

- Ahora ejecute el script así

```
[root@servidorA migration]# ./migrate_base.pl > ~/base.ldif
```

Este comando creará un archivo llamado **base.ldif** dentro de su directorio de inicio.

- Asegúrese de que **slapd** esté funcionando y entonces importe las entradas en el archivo **base.ldif** en el directorio OpenLDAP. Escriba

```
[root@servidorA ~]# ldapadd -c -x -D "cn=manager,dc=ejemplo,dc=org" \
-w -f base.ldif
```

5. Ahora necesitamos exportar los usuarios actuales del sistema en el archivo `/etc/passwd` hacia un archivo tipo LDIF. Utilicemos el script `/usr/share/openldap/migration/migrate_passwd.pl`. Escriba

```
[root@servidorA migration]# ./migrate_passwd.pl /etc/passwd > \
~/ldap-users.ldif
```

6. Ahora podemos iniciar la importación de todas las entradas de usuarios en el archivo `ldap-users.ldif` hacia nuestra base de datos OpenLDAP. Utilizaremos el comando `ldapadd`. Escriba

```
[root@servidorA ~]# ldapadd -x -D "cn=manager,dc=ejemplo,dc=org" -W -f \
ldap-users.ldif
```

7. Proporcione la contraseña rootdn cuando se le pregunte.

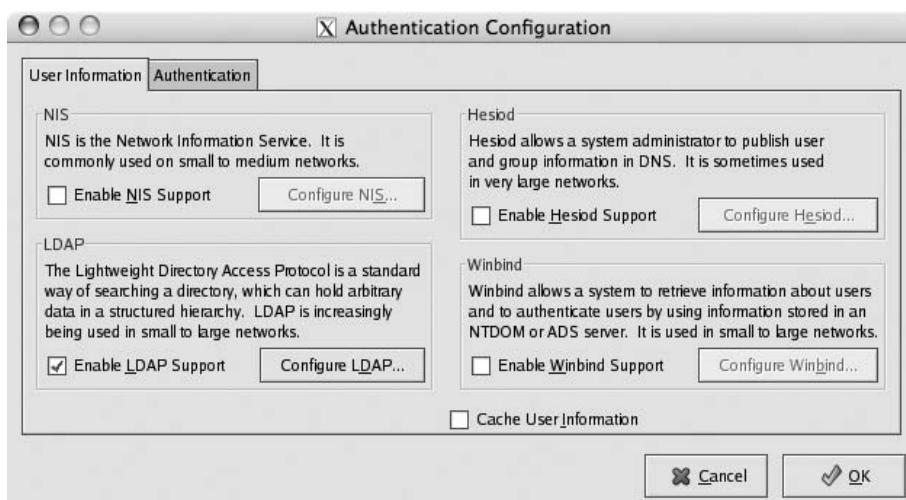
Configuración del cliente

La configuración de un sistema cliente para que utilice el directorio LDAP para autenticación es muy fácil en sistemas Fedora o RHEL. Fedora Core tiene una herramienta GUI (**system-config-authentication**) que facilita el procedimiento.

1. Inicie la herramienta desde una línea de comandos. Escriba

```
[root@servidorB ~]# system-config-authentication
```

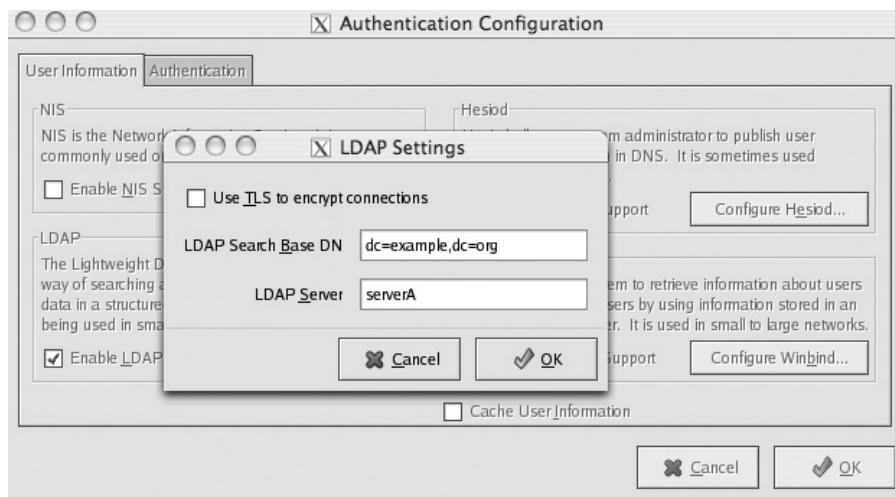
Aparecerá una ventana similar a la que mostramos enseguida.



2. En la ventana Authentication Configuration active la opción Enable LDAP Support.

3. Ahora haga clic en el botón Configure LDAP.

Aparecerá una ventana similar a la que mostramos enseguida.



4. Introduzca la información apropiada para su ambiente. En esta ventana, en el campo LDAP Search Base DN, especificamos “dc=ejemplo,dc=org”, sin las comillas; y en el campo LDAP Server ponemos “servidorA”, sin las comillas (también puede especificar la dirección IP).
5. Haga clic en OK.

Acabamos de describir una manera muy sencilla de habilitar un sistema cliente Fedora Core que utiliza un servidor OpenLDAP para la autenticación de usuarios. No nos ocupamos de otros detalles porque sólo se trata de una prueba de concepto. Sin embargo, aquí están otros detalles con los que quizás deba lidiar bajo condiciones de operación real:

- ▼ **Directorios de inicio** Quizá tenga que asegurarse de que los directorios de inicio de los usuarios están disponibles para ellos cuando se conectan desde cualquier sistema. Una posible forma de hacerlo sería compartir dichos directorios mediante NFS exportando los recursos compartidos a todos los sistemas clientes.
- **Seguridad** Nuestro sistema de muestra no tenía medidas de seguridad interconstruidas en la infraestructura. Ello debe ser de importancia suprema bajo condiciones de operación reales; de manera que las contraseñas de usuarios no salgan volando a través de la red como texto plano.
- ▲ **Otros asuntos** Hay otros asuntos que no tratamos aquí pero es nuestra intención dejarlos como ejercicio mental para que se tropiece con ellos y enriquezca su experiencia.

RESUMEN

En este capítulo tratamos algunas nociones básicas sobre LDAP. En su mayor parte nos concentramos en la implementación de Código Fuente Abierto conocida como OpenLDAP. Discutimos los componentes de OpenLDAP, los demonios que van del lado del servidor y las herramientas que van del lado del cliente, utilizadas para consultar y modificar la información almacenada en un directorio LDAP. Creamos un directorio muy sencillo y lo poblamos con algunas entradas de ejemplo.

Apenas si tocamos la superficie sobre el tema. LDAP es materia muy amplia como para hacerle justicia en un solo capítulo. Pero esperamos haber estimulado su apetito y haberlo puesto en el camino correcto con algunas ideas y conceptos esenciales.

CAPÍTULO 26



Servicios
de impresión

Típicamente, la impresión con Linux o UNIX nunca ha sido un proceso claro y directo. Sin embargo, con la llegada del Common UNIX Printing System (CUPS) (Sistema de impresión común de UNIX), la impresión en Linux se hizo más fácil en su configuración y uso. Anteriormente las impresoras para las que se ofrecía mayor soporte eran impresoras PostScript, tanto de Hewlett-Packard como de otros fabricantes. Y conforme Linux ha ido demostrando su viabilidad como una estación de trabajo para el escritorio, se ha hecho necesaria una mejor solución para servicios de impresión. Esa solución es CUPS. Este capítulo cubrirá la instalación del sistema CUPS, al igual que las tareas administrativas involucradas con el mantenimiento de un ambiente para servicios de impresión.

TERMINOLOGÍA DE IMPRESIÓN

Hay varios sistemas de impresión disponibles en el mundo de Linux, todos basados en mayor o menor medida en el venerable sistema de impresión BSD. Enseguida mostramos algunos términos con los que deberá familiarizarse:

- ▼ **Impresora** Un dispositivo periférico que, por lo general, se conecta a una computadora anfitrión o a la red.
- **Trabajos de impresión** El archivo o conjunto de archivos que se envían a imprimir se conoce como trabajo de impresión.
- **Administrador de impresión** Es un software que se encarga de administrar los trabajos de impresión. Es responsable de recibir trabajos de impresión, almacenarlos, organizarlos en una cola de trabajos y enviarlos al hardware físico que se encarga de imprimirlós. Son programas que siempre están en memoria (demonios), atentos a las solicitudes de impresión, y por ello es usual que se les refiera como "servidores de impresión". Los siguientes programas son ejemplos de administradores de impresión (spoolers):
 - ▼ **LPD** Este es el auténtico y original programa BSD Line Printer Daemon (Demónio de impresora en línea). Es un sistema de impresión veterano.
 - **LPRng** Esta es una implementación mejorada, extendida y portátil de la funcionalidad que ofrece el administrador Berkeley **lpr**. Combina las mejores características del sistema de impresión de System V con las funciones del sistema Berkeley.
 - ▲ **CUPS** Proporciona una capa de impresión portátil para sistemas basados en UNIX. Utiliza el Internet Printing Protocol (IPP) (Protocolo de impresión por Internet) como la base para efectuar la administración de trabajos y colas de impresión.
- **PDL** Lenguaje para descripción de páginas (*page description language*). Las impresoras aceptan la llegada de trabajos en este formato. PostScript y PCL son ejemplos de PDL.
- **PostScript** Los archivos PostScript en realidad son programas. Es un lenguaje de programación basado en capas. La mayoría de los programas UNIX/Linux generan salida para imprimir en este formato. Las impresoras basadas en este lenguaje ofrecen soporte directo al formato.
- **Ghostscript** Un intérprete de PostScript basado en software para impresoras que no son compatibles con este formato. Se utiliza para impresiones controladas por software. Este intérprete genera el mismo lenguaje para una impresora en particular a partir de una salida en formato PostScript. Ejemplos son Aladdin Ghostscript (versión comercial), GNU Ghostscript (versión gratuita) y ESP Ghostscript (CUPS).
- ▲ **Filtro** Los filtros son programas especiales o scripts que procesan datos (trabajos de impresión) antes de que éstos sean enviados a la impresora. Los administradores de impresión

envían los trabajos a los filtros y entonces éstos los pasan a la impresora. La traducción y la contabilidad del formato del archivo por lo general se llevan a cabo en la capa de filtrado.

EL SISTEMA CUPS

CUPS está ganando amplia aceptación en toda la comunidad Linux/UNIX. Hasta la nueva versión para OS X de Apple ofrece soporte para CUPS. Lo que eso significa es que ahora puede tener un ambiente de impresión ubicuo sin importar el sistema operativo que utilice. Junto con LPR, el protocolo de impresión estándar de UNIX, CUPS respalda servicios de impresión Samba y el nuevo Protocolo de Impresión para Internet. Utilizando el concepto de *clases* de impresión, el sistema CUPS imprimirá un documento en un grupo de impresoras para sacar provecho de un grupo de impresoras en ambientes con altos volúmenes de impresión. Puede actuar como un administrador de impresión central o sólo proveer un método de impresión para su impresora local.

Funcionamiento de CUPS

Esta sección cubre el proceso de instalación de CUPS, así como el control del servicio.

Easy Software Products creó el software CUPS y está disponible en <http://www.easysw.com/>. Existen dos métodos para instalar CUPS en su sistema: mediante la distribución de Linux que ya utiliza o mediante la compilación del código fuente. El primer método es el más recomendado, ya que las distribuciones incluyen típicamente soporte para las impresoras más populares, listo para funcionar con CUPS. Cuando lo compila a mano, deberá conseguir por su cuenta los controladores de sus impresoras.

Instalación de CUPS

Como ha sucedido con la mayor parte del software con el que hemos lidiado hasta ahora, el software CUPS está disponible en dos formas: el código fuente, desde el cual puede construir una compilación del software, y los binarios RPM precompilados.

Si necesita compilar CUPS desde la fuente, siga las instrucciones que vienen con el paquete del software. Puede encontrar el código fuente de CUPS en <http://www.cups.org/>. Las instrucciones de instalación acompañan al software. También querrá dar un vistazo al paquete Foomatic ubicado en <http://www.linuxprinting.org>. Este sitio proporciona una gran gama de controladores de impresora para diversos sistemas de impresión, incluido CUPS.

Si tiene una distribución Linux como Fedora Core, RHEL, SuSE, cAos o Mandrake, CUPS debe estar disponible como un paquete RPM; de hecho, CUPS es el sistema de impresión predeterminado que se utiliza en esas distribuciones.

Aunque propiamente no sea un método de instalación, se recomienda apegarse a la versión de CUPS que acompaña a su distribución. El vendedor de esa distribución ha hecho un gran esfuerzo para asegurar que CUPS trabaje bien con su sistema. Si está en el caso de los desafortunados que no tienen una distribución Linux que venga con CUPS integrado de fábrica, puede compilar el paquete a partir del código fuente (vea la siguiente sección).

Debido a que la mayoría de los sistemas de Linux ya tienen instalado CUPS desde la instalación inicial del sistema operativo, primero debe preguntar a la base de datos RPM de su sistema si ya está instalado dicho software. Escriba

```
[root@servidorA ~]# rpm -q cups  
cups-1.1.23-15
```

Si esta consulta no regresa nada, puede instalar CUPS rápidamente en sistemas Fedora Core o RHEL si escribe

```
[root@servidorA ~]# up2date -i cups
```

o bien

```
[root@servidorA ~]# yum install cups
```

En un sistema SuSE debería serle posible ejecutar el siguiente comando para llevar a cabo la instalación de CUPS:

```
[root@servidorA ~]# yast -i cups
```

Una vez instalado, necesitará activar el demonio CUPS. En un sistema Fedora haría lo siguiente:

```
[root@servidorA ~]# service cups restart
```

En otro tipo de sistemas debería serle posible iniciar CUPS al ejecutar el script de arranque en forma directa, como sigue:

```
[root@servidorA ~]# /etc/init.d/cups start
```

Ello iniciará el sistema de impresión CUPS y le permitirá conectarse a la interfaz Web para añadir impresoras.

Configuración de CUPS

El archivo principal de configuración para el demonio de impresión CUPS se llama **cupsd.conf**. Lo normal es que este archivo esté ubicado en el directorio **/etc/cups/**. Es un archivo de texto plano con directrices (sintaxis) similares al archivo respectivo del servidor Web Apache. Las directrices determinan cómo operará el servidor.

El archivo está bien comentado y prácticamente lo único que necesita hacer es quitar o poner líneas de comentario en el archivo a fin de activar o desactivar las funciones que requiera.

Algunas directrices interesantes utilizadas en el archivo **cupsd.conf** son:

- ▼ **Browsing** Controla si es o no posible utilizar un navegador Web para administrar las funciones de impresión en red.
- **BrowseProtocols** Especifica los protocolos que se utilizarán cuando se recolecten o distribuyan impresoras compartidas en una red local.
- **BrowseInterval** Especifica la máxima cantidad de tiempo que puede transcurrir entre actualizaciones de navegación.
- **BrowseAddress** Especifica la dirección a la cual se enviará información de navegación.
- **Location** Especifica el control de acceso y las opciones de autenticación para la ruta o el recurso HTTP especificado.
- ▲ **ServerName** Especifica el nombre de anfitrión que será reportado a los clientes.

Un aspecto interesante es la ubicación raíz, representada por la diagonal invertida (/). Dentro del archivo predeterminado **cupsd.conf**, la especificación de la ubicación raíz se ve como sigue (note que se han añadido números al inicio de las líneas a fin de facilitar su legibilidad):

- 1) <Location />
- 2) Order Deny,Allow
- 3) Deny From All
- 4) Allow From 127.0.0.1
- 5) </Location>

- ▼ **Línea 1** Este es el inicio de la directriz Location; aquí se define el inicio de “/”, que es la ruta de TODAS las operaciones get (get-printers, get-jobs, etc.), es decir, define el nivel superior del servidor Web.
- **Línea 2** Esta es la directriz de orden (Order). Define el control de acceso predeterminado para la ubicación en cuestión. Enseguida listamos los posibles valores de la directriz Order:
 - ▼ **Deny,Allow** Acepta solicitudes *sólo* de los anfitriones especificados en la directriz Allow.
 - ▲ **Allow,Deny** Acepta solicitudes de todos los sistemas, excepto de aquellos especificados en la directriz Deny.
- **Línea 3** La directriz Deny especifica los anfitriones a los que se les negará el acceso. En este caso la palabra reservada “All” significa todos los anfitriones.
- **Línea 4** La directriz Allow especifica los anfitriones a los que se les permitirá acceso. En este caso el único anfitrión permitido es el anfitrión localhost, es decir, la dirección de retorno (127.0.0.1).
- ▲ **Línea 5** Esta es la etiqueta de cierre de la directriz Location.

SUGERENCIA El comportamiento predeterminado de CUPS sólo permite acceso desde el anfitrión local. Para cambiarlo sólo tendría que cambiar la directriz Allow de “Allow From 127.0.0.1” a “Allow From All” y luego cancelar como comentario la directriz “Deny From All”.

AGREGADO DE IMPRESORAS

El primer paso después de haber compilado y activado el sistema CUPS es conectarse a la interfaz Web. Dicha interfaz está disponible a través del puerto 631. En su navegador Web sólo tiene que escribir <http://localhost:631>. En forma predeterminada, debe estar registrado en el mismo servidor que está tratando de administrar. Lo interesante aquí es que el puerto 631 es el mismo puerto por el que CUPS acepta trabajos de impresión. Cuando se conecta a la página Web verá una página similar a la de la figura 26-1.

NOTA Si quiere administrar impresoras desde otras ubicaciones que no sean el servidor desde el cual está trabajando, necesitará modificar el archivo **cupsd.conf** para permitir que otros anfitriones también se conecten. Es usual que este archivo resida en el directorio **/etc/cups/**.

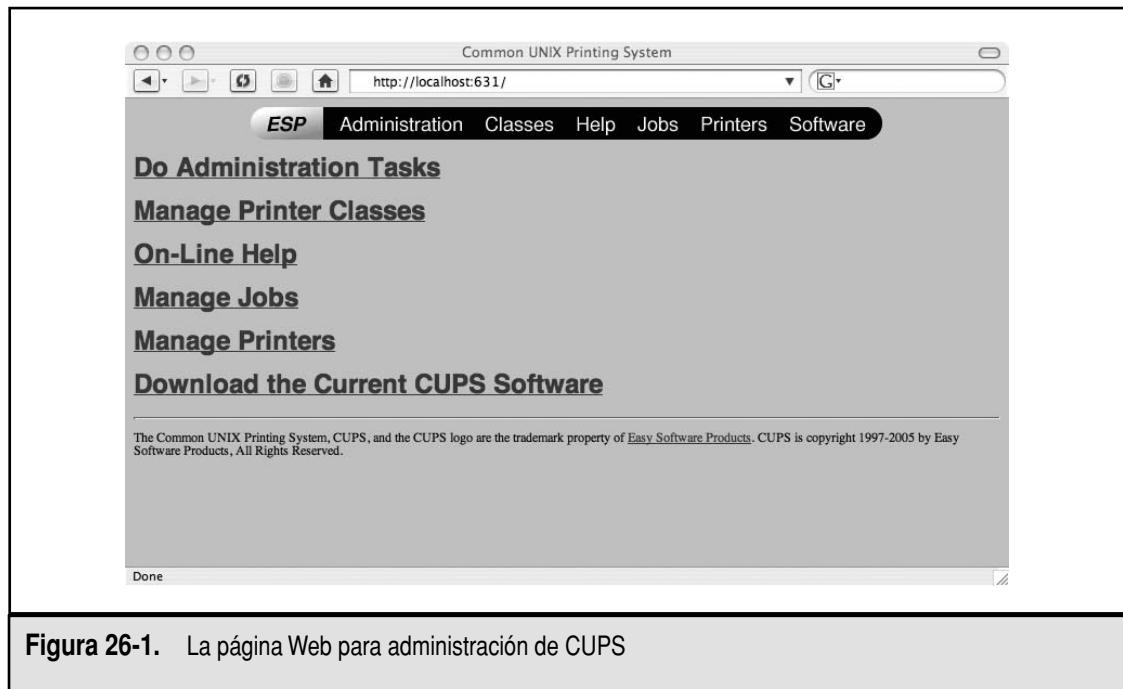


Figura 26-1. La página Web para administración de CUPS

Impresoras locales e impresoras remotas

El procedimiento para añadir impresoras en CUPS es muy sencillo. Un importante dato que necesitará es cómo está conectada la impresora a su sistema. La conexión entre un anfitrión y una impresora puede realizarse eligiendo uno de dos posibles métodos: de manera directa (local) o a través de la red. Existen varios modos o posibilidades para cada método. Los modos mediante los cuales CUPS se dirige a los recursos de impresión se especifican utilizando lo que se conoce como *Uniform Resource Information (URI)* (*Información de recursos uniformes*). Enseguida mostramos los posibles dispositivos URI que se pueden configurar en CUPS:

- ▼ **Directamente conectados (local)** Es casi seguro que un sistema casero que ejecuta Linux estará conectado en forma directa a la impresora a través de un cable que se conecta al puerto paralelo, al puerto serial o al puerto USB. Esto es un ejemplo de impresora conectada en forma directa. En la jerga de CUPS, algunos dispositivos URI posibles para impresoras conectadas en forma local se especificarán como sigue:
 - ▼ **parallel:/dev/lp*** Para una impresora conectada al puerto paralelo
 - **serial:/dev/ttyS*** Para una impresora conectada al puerto serial
 - ▲ **usb:/dev/usb/lp*** Para una impresora conectada al puerto USB
- **IPP (red)** Siglas para Internet Printing Protocol (Protocolo de impresión por Internet). Permite que una impresora se pueda usar a través de Internet utilizando IPP. La mayoría de los sistemas operativos modernos ofrecen soporte para este protocolo y, por

ende, ello no representa mayor problema. Un ejemplo de URI para un dispositivo IPP en CUPS es `ipp://nombreanfitrion/ipp/`.

- **LPD (red)** Siglas para Line Printer Daemon (Demonio de impresora en línea). CUPS ofrece soporte para impresoras que están conectadas a sistemas que corren este demonio. De igual forma, la mayoría de los sistemas Linux/UNIX (e inclusive algunos servidores Windows) lo soportan. De manera que si una impresora está conectada a un anfitrión que soporta LPD, puede utilizar CUPS para compartir esa impresora mediante la red con otros anfitriones que no necesariamente soportan LPD. Casi todas las impresoras láser de HP que incorporan funciones de conectividad en red soportan de fábrica LPD.
Un ejemplo de URI para dirigirse a una impresora LPD es `lpd://nombreanfitrion/cola`, donde *nombreanfitrion* es el nombre de la máquina donde está funcionando LPD.
- **SMB (red)** Siglas para Service Message Block (Servicio de bloque de mensajes). Es la base para compartir archivos e impresoras en redes Windows. Los anfitriones Linux/UNIX también ofrecen soporte para SMB mediante el uso del software Samba. Por ejemplo, si un sistema Windows (o un servidor Samba) tiene instalada una impresora compartida, CUPS puede configurarse para acceder y ofrecer dicha impresora entre sus clientes. Un ejemplo de URI para dirigirse a una impresora compartida SMB es `smb://nombreservidor/recursocompartido`, donde *recursocompartido* es el nombre con el cual la impresora ha sido compartida en el sistema Windows o en el servidor Samba.
- ▲ **Impresora compartida en red** ¿En serio? Se refiere a una clase de impresoras que tienen características de red interconstruidas. Estas impresoras no necesitan estar conectadas a un sistema independiente. Es usual que tengan algún tipo de interfaz de red propia, ya sea Ethernet, inalámbrica, o algún otro método para conectarse en forma directa a una red. Un tipo de impresoras muy popular en esta categoría es la serie HP Jetdirect. Un ejemplo de URI para dirigirse a una impresora como tal es `socket://dirección_ip:puerto`, donde *dirección_ip* es la dirección IP de la impresora y *puerto* es, como su nombre lo indica, el número de puerto en el que la impresora escucha solicitudes de impresión. Es usual que dicho puerto sea 9100 en las impresoras HP Jetdirect.

Uso de la interfaz Web

Existen dos maneras de añadir y configurar impresoras en CUPS. Una es mediante la interfaz Web, utilizando algún tipo de navegador; la otra es mediante órdenes proporcionadas desde una línea de comandos. El primer método es quizás el más fácil debido a que utiliza un asistente que lo orienta y lo acompaña a través de todo el proceso.

Esta sección lo llevará por todos los pasos a efecto de poner en marcha una impresora mediante la interfaz Web de CUPS. Configuraremos una impresora imaginaria con las siguientes propiedades:

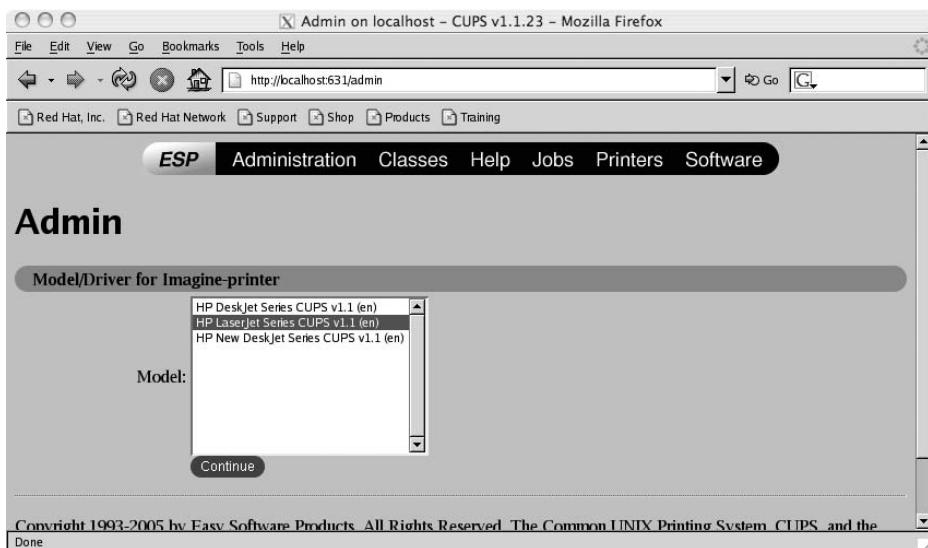
Name:	Imagine-printer
Location:	Building 3
Description:	You only need to imagine to print here.
Connection Type:	Local. Connected to Parallel port
Make:	HP
Model:	LaserJet Series

Demos inicio al proceso.

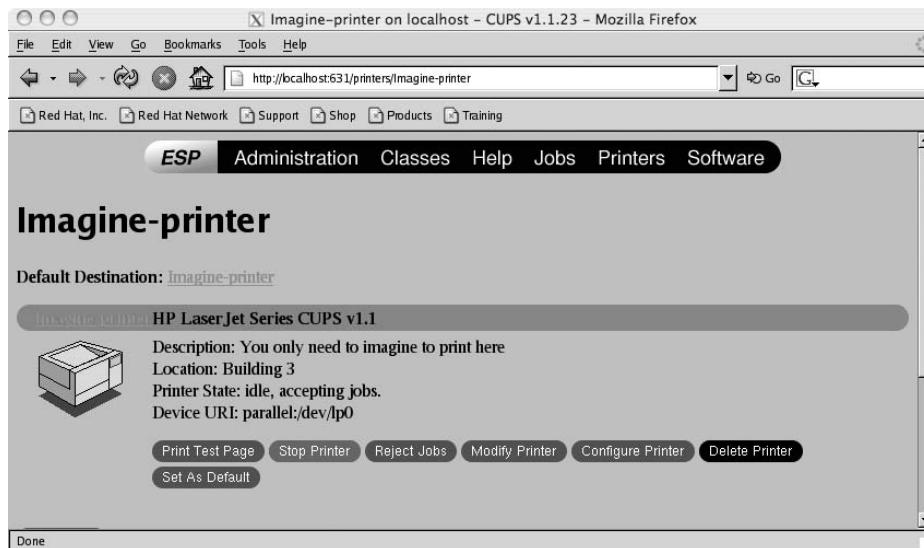
1. Si ya tiene abierta una sesión de trabajo en el sistema que está ejecutando CUPS, inicie un navegador Web y diríjalo al siguiente URL:
`http://localhost:631`
Cuando se le pregunte el nombre del usuario y la contraseña escriba **root** como el nombre del usuario y proporcione la contraseña que corresponda.
2. Después de registrarse haga clic en el vínculo Do Administration Tasks.
3. En la parte inferior de esa página haga clic en el vínculo Add Printer.
4. En la página Add Printer proporcione la información para la impresora que se le dio en la página anterior; es decir, los campos para Name, Location y demás.
5. Haga clic en el botón Continue cuando termine de introducir los datos.
6. En la siguiente página utilice la lista para seleccionar la opción “parallel port #1” y haga clic en Continue.
7. En la página Make/Model seleccione la marca de la impresora (HP para este ejemplo) y haga clic en Continue.

SUGERENCIA Es obvio que las opciones existentes para los fabricantes de impresoras mostrados en la lista no cubren todas las marcas existentes. Si necesita mayor variedad, en sistemas Fedora o RHEL puede instalar el paquete RPM print-cups. Dicho paquete proporciona controladores adicionales para varios fabricantes de impresoras además de los que aparecen de fábrica con la versión básica del software de fuente abierta CUPS (no olvide que CUPS también está disponible en versión comercial).

8. Ahora seleccionará el modelo de la impresora de la opción Model/Driver. Seleccione la opción HP Laserjet Series de la lista de modelos mostrada y haga clic en Continue.



9. Aparecerá una página que le confirmará si la impresora se añadió con éxito. Haga clic en el nombre de la impresora (Imagine-printer).
10. Verá una página similar a la mostrada enseguida. La página mostrará las propiedades de la impresora que acaba de añadir.



Observe que el software generó en automático el dispositivo URI apropiado (parallel: //dev/ lp0) con el cual se comunicará con la impresora.

Ahora póngase su mejor sombrero imaginativo e imagine que hace clic en el vínculo “Print test page” en la página de propiedades de la impresora que acaba de añadir. Enseguida imagine que una página de prueba se imprime con éxito.

Para agregar una impresora desde una línea de comandos

El uso de la línea de comandos es el segundo método para añadir impresoras al sistema CUPS. Una vez que se sienta cómodo con la forma como trabaja CUPS, quizás encuentre que administrarlo resulta más rápido si lo hace desde una línea de comandos. Para añadir una impresora con este método necesitará definir cierta información pertinente tal como el nombre, el controlador y URI.

Esta sección abordará a detalle un sencillo ejemplo a efecto de poner en marcha una impresora utilizando las herramientas CUPS desde la línea de comandos. Al igual que en el ejemplo anterior, configuraremos una impresora imaginaria. La nueva impresora tendrá las mismas propiedades que la anterior, a excepción del nombre de la impresora (también llamado *cola de impresión*). Llamaremos a esta otra impresora “Imagine-printer-number-2”. En vez del puerto paralelo que utilizamos en el ejemplo anterior, utilizaremos un dispositivo URI distinto para que CUPS se comunique con la impresora. Esta vez supondremos que se trata de una impresora compartida en red con la dirección IP 192.168.1.200 y que escuchará en el puerto 9100; esto es, el dispositivo URI será socket://192.168.1.200:9100.

- Si ya tiene abierta una sesión de trabajo en el sistema (como el usuario root), abra una terminal virtual y liste las colas de impresión que actualmente tiene configuradas. Para su sistema use la utilidad **lpstat**. Escriba

```
[root@servidorA ~]# lpstat -a
Imagine-printer accepting requests since Jan 01 00:00
```

- Ahora utilice el comando **lpadmin** para añadir la impresora. Este comando es extenso debido a todas las opciones que lo componen, así que verá varias líneas en este listado. Escriba

```
[root@servidorA ~]# lpadmin -p "Imagine-printer-number-2" -E \
-v socket://192.168.1.200 \
-P /usr/share/cups/model/laserjet.ppd.gz \
-D "You only need to imagine to print here" \
-L "Building 3"
```

- Utilice el comando **lpstat** de nuevo para listar todas las impresoras que están dadas de alta en su sistema. Escriba

```
[root@servidorA ~]# lpstat -a
Imagine-printer accepting requests since Jan 01 00:00
Imagine-printer-number-2 accepting requests since Jan 01 00:00
```

- También puede ver la impresora que acaba de añadir en la interfaz Web de CUPS. Lleve a su navegador Web al siguiente URL:

<http://localhost:631/printers>

Verá una página similar a la mostrada en la figura 26-2.

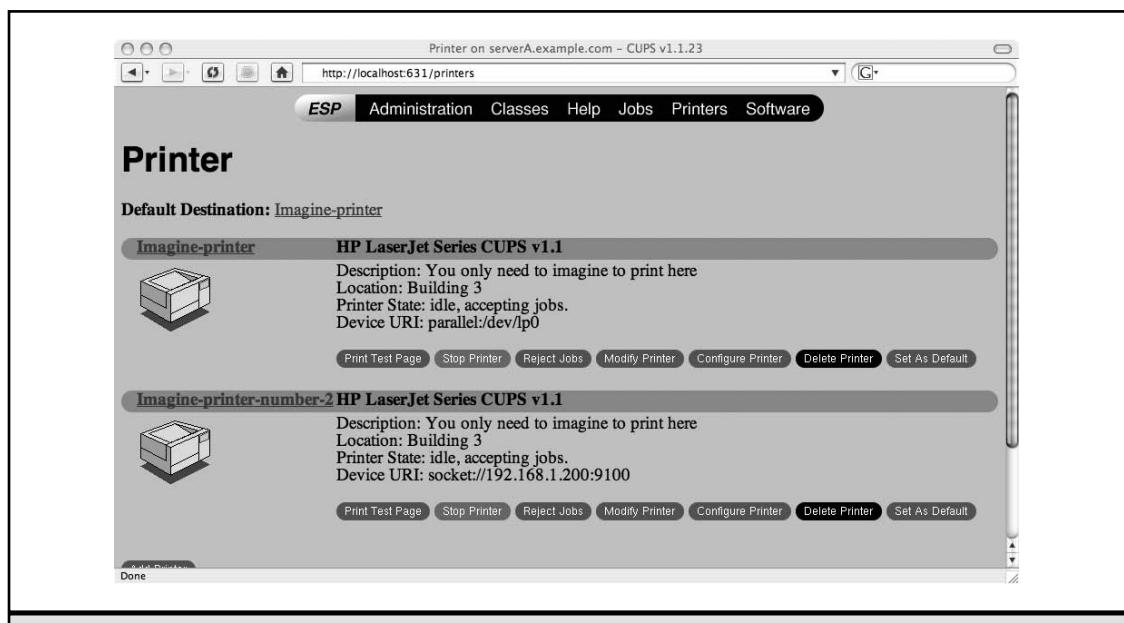


Figura 26-2. La página “Printers” (Impresoras) en la página de administración Web de CUPS

ADMINISTRACIÓN DE RUTINA DE CUPS

La puesta en marcha de una impresora es la mitad de la batalla en la administración de ambientes de impresión. Esperamos que la sección anterior le haya dado suficiente información para ponerlo al día en esa materia. Esta sección abordará algunos de los procedimientos de rutina para la administración de ambientes de impresión, verá tareas como eliminación de impresoras, manejo de la cola de impresión y visualización del estado de trabajos de impresión. Para llevar a cabo estas tareas utilizaremos ambas líneas de comandos e interfaz Web.

Configuración de la impresora predeterminada

En un sistema con múltiples colas de impresión puestas en marcha, tal vez sea deseable configurar alguna de ellas como el dispositivo predeterminado donde se imprimirán los trabajos enviados por los clientes en aquellos casos en los que el nombre de la impresora no se especifica explícitamente.

Por ejemplo, para establecer la impresora llamada “Imagine-printer-number-3” como la impresora predeterminada, escriba

```
[root@servidorA ~]# lpadmin -d imagine-printer-number-3
```

Activación y desactivación de impresoras

Desactivar una impresora guarda cierto parecido con ponerla “fuera de línea”. En este estado la cola de impresión puede seguir aceptando trabajos, pero en realidad no los imprimirá. Los trabajos de impresión se irán acumulando hasta que la impresora se active o se reinicie. Ello es útil cuando en alguna situación el dispositivo periférico sufre alguna falla física que le impide seguir operando momentáneamente y el administrador no quiere interrumpir el envío de trabajos por parte de los usuarios.

Para desactivar una impresora llamada “imagine-printer-number-3”, escriba

```
[root@servidorA ~]# /usr/bin/disable imagine-printer-number-3
```

Para activar esa misma impresora, escriba

```
[root@servidorA ~]# /usr/bin/enable imagine-printer-number-3
```

Aceptación y rechazo de trabajos de impresión

Puede configurar cualquier impresora administrada por CUPS para que acepte o rechace trabajos de impresión. Ello es distinto a desactivarla, pues una impresora que se configura para rechazar trabajos de impresión sencillamente *no* aceptará ninguna solicitud. Hacer que una impresora rechace trabajos es útil para situaciones en las que es necesario quitar la impresora de servicio por un periodo prolongado sin que sea necesario eliminarla por completo.

Cuando establece que una impresora debe rechazar trabajos de impresión, primero completará los trabajos que tenga pendientes e inmediatamente dejará de aceptar nuevas solicitudes.

A manera de ejemplo, para hacer que una impresora llamada “imagine-printer-number-3” rechace trabajos de impresión, escriba

```
[root@servidorA ~]# /usr/sbin/reject imagine-printer-number-3
```

Utilice el comando **lpstat** para ver el estado de esta impresora.

```
[root@servidorA ~]# lpstat -a imagine-printer-number-3
Imagine-printer-number-3 not accepting requests since Jan 01 00:00 -
    Rejecting Jobs
```

Para hacer que la impresora “imagine-printer-number-3” vuelva a aceptar trabajos de impresión, escriba

```
[root@servidorA ~]# /usr/sbin/accept imagine-printer-number-3
```

Visualice una vez más el estado de la impresora. Escriba

```
[root@servidorA ~]# lpstat -a imagine-printer-number-3
Imagine-printer-number-3 accepting requests since Jan 01 00:00
```

Gestión de privilegios de impresión

En su estado de instalación recién terminada, todos los usuarios pueden enviar trabajos de impresión a cualquier impresora administrada por CUPS. En ambientes multiusuario de gran envergadura quizás encuentre necesario controlar cuáles usuarios o grupos tienen acceso a tales o cuales impresoras. Esto puede suceder por razones de seguridad o meramente por disposiciones o políticas internas. CUPS ofrece una sencilla forma para hacerlo mediante el uso de la utilidad **lpadmin**.

Por ejemplo, a efecto de permitir que sólo los usuarios llamados *fernando* y *oscar* puedan imprimir en la impresora llamada “imagine-printer”, escriba

```
[root@servidorA ~]# lpadmin -p imagine-printer -u allow:fernando,oscar
```

Para llevar a cabo la intención opuesta, denegando el acceso a la misma impresora para los usuarios *fernando* y *oscar*, escriba

```
[root@servidorA ~]# lpadmin -p imagine-printer -u deny:fernando,oscar
```

Para retirar todas las restricciones anteriores y permitir que todos los usuarios puedan imprimir en la impresora llamada “imagine-printer”, escriba

```
[root@servidorA ~]# lpadmin -p imagine-printer -u allow:all
```

Eliminación de impresoras

Para eliminar una impresora llamada “bad printer”, desde la línea de comando, escriba

```
[root@servidorA ~]# lpadmin -x bad-printer
```

Gestión de impresoras mediante la interfaz Web

Casi todos los procedimientos anteriores pueden realizarse desde la interfaz Web de CUPS. Utilizando botones y vínculos puede eliminar impresoras con facilidad, controlar trabajos de impresión, modificar las propiedades de una impresora, detener impresoras, rechazar trabajos de impresión y demás.

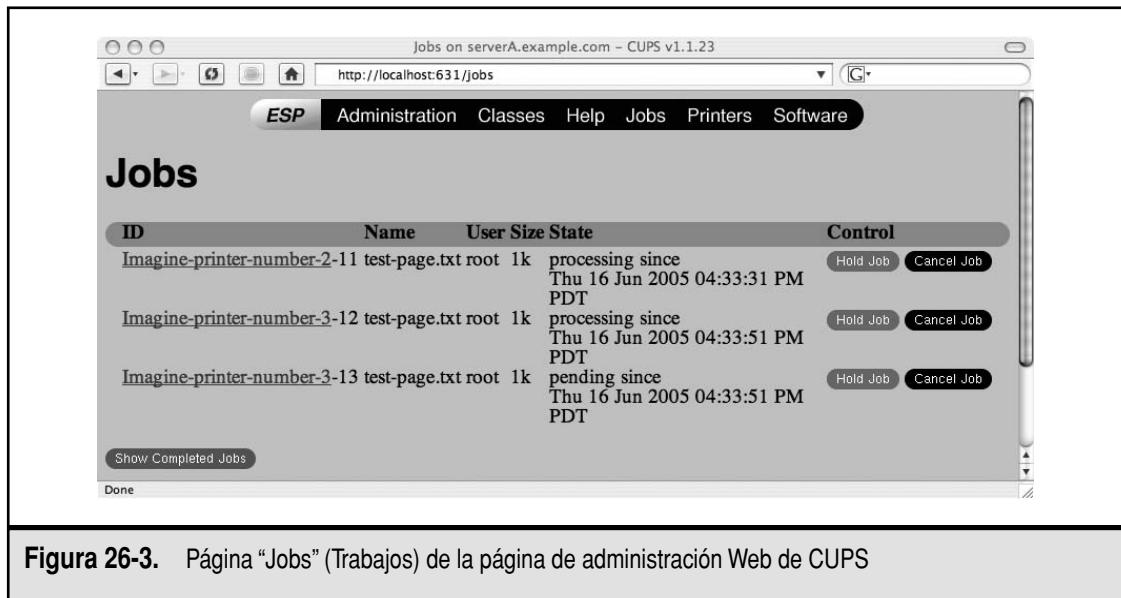


Figura 26-3. Página “Jobs” (Trabajos) de la página de administración Web de CUPS

Un ejemplo de su uso es el siguiente. Como administrador, seguro necesitará verificar las colas de impresión con cierta regularidad para asegurarse de que todo marcha como debe. Al hacer clic en el botón Jobs (o al ir directamente a <http://localhost:631/jobs>) en la interfaz Web, traerá una página similar a la mostrada en la figura 26-3. Como puede ver, si existen trabajos de impresión en la cola, tiene tres opciones a escoger; de lo contrario, sólo podrá mostrar los trabajos terminados haciendo clic con el botón Show Completed Jobs. Puede detener (poner en pausa) el trabajo o cancelarlo por completo.

También puede llevar a cabo toda una gama de tareas administrativas si navega hasta la página Admin (Administración) de CUPS. En su sistema local, el URL para esta página es <http://localhost:631/printers/>.

USO DE HERRAMIENTAS DE IMPRESIÓN DEL LADO DEL CLIENTE

Después de haber visto diversos aspectos de la instalación y administración de CUPS, es tiempo de analizar cómo se imprime con el sistema Linux.

Cuando una máquina cliente manda imprimir un documento, la solicitud se envía al servidor de impresión, se asigna ordenadamente en una cola y se genera un trabajo o cola de impresión. Un trabajo de impresión puede estar en uno de dos estados: *en progreso* o *en pausa*. Cuando un administrador pone el trabajo en pausa se dice que entró *manualmente* en ese estado. Por el contrario, cuando se le acaba el papel a la impresora, el trabajo de impresión se pone en pausa *en forma automática*. Cuando algo sale realmente mal en la impresora, los trabajos de impresión pueden acumularse y causar un problema cuando la impresora regresa en línea.

En esta sección veremos algunos comandos que pueden utilizarse para imprimir, así como algunos otros que se usan para administrar colas de impresión. Cubrimos estos asuntos desde el punto de vista del usuario y su interacción con el sistema de impresión.

lpr

El usuario usa el comando **lpr** para imprimir documentos. La mayoría de los documentos PostScript y los documentos de texto pueden imprimirse en forma directa utilizando el comando **lpr**. Si está utilizando AbiWord o StarOffice tendrá que configurar esas aplicaciones para imprimir en el dispositivo correcto.

Empezaremos por crear un archivo de texto plano que intentaremos imprimir. El archivo contendrá un texto corto como "Cuitzeo" y será almacenado como **michoacan.txt**.

- Escriba lo siguiente:

```
[root@servidorA ~]# echo "Cuitzeo" >> michoacan.txt
```

- Encuentre el nombre de la impresora predeterminada configurada en el sistema. Escriba

```
[root@servidorA ~]# lpstat -d
system default destination: Imagine-printer
```

- Envíe el archivo **michoacan.txt** a la impresora predeterminada. Escriba

```
[root@servidorA ~]# lpr michoacan.txt
```

Esto imprimirá el documento **michoacan.txt** en la impresora predeterminada, la cual es típicamente la primera que se instaló.

- Ahora envíe el mismo documento a otra impresora imaginaria instalada páginas atrás, misma que se llamaba "imagine-printer-number-2". Escriba

```
[root@servidorA ~]# lpr -P Imagine-printer-number-2 michoacan.txt
```

Una vez que haya ejecutado este comando, en breve la impresora deberá empezar a imprimir, a menos que esté imprimiendo un archivo muy grande.

- Para ver el estado de este trabajo de impresión utilice el comando **lpq**.

lpq

Después de haber enviado el trabajo puede ver lo que está sucediendo en la cola de impresión si utiliza el comando **lpq**. Si acaba de imprimir un trabajo y nota que no sale de la impresora, utilice este comando para mostrar la lista actual de trabajos en la cola de impresión asignada a esa impresora. Es común que vea varios trabajos de impresión en la cola y, después de cierta investigación, descubra que la impresora se quedó sin papel. Si necesita quitar el trabajo de la cola de impresión puede utilizar el comando **lprm** que explicamos en la siguiente sección.

Por ejemplo, para ver el estado de la solicitud enviada a la impresora predeterminada, escriba

```
[root@servidorA ~]# lpq -av
Rank      Owner     Job      File(s)                      Total Size
active    root       2        michoacan.txt                  1024 bytes
```

Para ver el estado del trabajo de impresión enviado a la otra impresora, escriba

```
[root@servidorA ~]# lpq -av -P Imagine-printer-number-2
Imagine-printer-number-2 is ready and printing
Rank      Owner    Job      File(s)          Total Size
active    root      2       michoacan.txt      1024 bytes
```

Como se mostró en salidas anteriores, ambos trabajos están atorados en las colas de impresión imaginarias debido a que no utilizamos nuestra imaginación suficientemente bien. Será necesario que eliminemos los trabajos de impresión, como se verá enseguida.

lprm

Cuando se da cuenta de que en realidad no quería enviar el trabajo de impresión que acaba de mandar, quizás tenga suerte si lo elimina antes de que sea impreso. Para hacerlo utilice el comando **lprm**. Ello retirará el trabajo de la cola de impresión.

Por ejemplo, para eliminar el trabajo que tiene 2 como su ID (utilice el comando **lpq** para obtenerlo, como se recomendó en la sección anterior) en la impresora predeterminada, escriba

```
[root@servidorA ~]# lprm 2
```

Para eliminar un trabajo de una impresora específica sólo añada la opción **-P**. Por ejemplo, para eliminar el trabajo que tiene 2 como su ID en la impresora llamada "Imagine-printer-number-2", escriba

```
[root@servidorA ~]# lprm 2 -P imagine-printer-number-2
```

Si usted es el usuario raíz, puede purgar todos los trabajos de impresión de una impresora llamada "Imagine-printer" si emite el comando **lprm** como sigue:

```
[root@servidorA ~]# lprm -P imagine-printer -
```

El guión al final de la línea significa "todos los trabajos".

SUGERENCIA En general, los usuarios regulares sólo pueden gestionar sus propios trabajos de impresión; esto es, el usuario A no puede ir y borrar un trabajo enviado por el usuario B de la cola de impresión. Desde luego, el superusuario raíz sí puede controlar los trabajos de todos. Asimismo, es preciso que comprenda que sólo tiene una estrecha y reducida ventana de tiempo para eliminar el trabajo, misma que empieza a transcurrir desde que lo envía a la impresora hasta que es capaz de eliminarlo. Por lo tanto, quizás encuentre que las solicitudes del comando **lprm** fallan debido a que éste se emite en forma tardía. Ello suele dar por resultado un error como "**lprm: Unable to lprm job(s)!**" Además, debe tener presente que el uso del guión (-) para purgar todos los trabajos de la cola de impresión es de uso exclusivo del usuario raíz.

RESUMEN

Este capítulo cubrió el Common UNIX Printing System (CUPS) (Sistema de impresión común de UNIX). Vimos procedimientos sencillos de administración de impresoras como su adición desde

la interfaz Web de CUPS y desde la línea de comandos, la gestión de impresoras y el control de trabajos de impresión bajo Linux. Mediante ejemplos también abordamos el uso común de algunas herramientas que van del lado del cliente y que sirven para gestionar trabajos de impresión en Linux. También vimos algunas de las directrices de configuración que se usan en **cupsd.conf**, el archivo de configuración principal de CUPS.

Sin embargo, apenas nos fue posible tocar la superficie de las habilidades y características de CUPS. Afortunadamente el software es acompañado de una amplia documentación en línea, misma que recomendamos leer si tiene planes de utilizar CUPS ampliamente a efecto de poner en marcha y administrar impresoras en su ambiente. La misma documentación también está disponible en Internet, en la página de inicio de CUPS: <http://www.cups.org>.

Una vez que tengan instalados los servicios de impresión en su servidor Linux encontrará que hacen su trabajo muy bien y le permiten enfocarse en otros nuevos e interesantes retos. Los problemas que pueden surgir de ahí en adelante se deben en buena medida a problemas atribuibles a la impresora, tales como papel atascado, abusos por parte del usuario o políticas internas.

CAPÍTULO 27



DHCP

La configuración manual de un puñado de direcciones IP en varias computadoras es una tarea relativamente simple. Sin embargo, dicha tarea sería ardua y desafiante si se tratara de la configuración manual de direcciones IP en un departamento, un edificio o una empresa con cientos o quizás miles de sistemas heterogéneos.

El Dynamic Host Configuration Protocol (DHCP) (Protocolo de configuración dinámica de anfitriones) para clientes y servidores Linux puede asistirle con estas faenas. Bajo este esquema, la máquina cliente se configura para obtener su dirección IP desde la red. Cuando inicia el software cliente de DHCP, éste transmite una señal que solicita a la red que le asigne una dirección IP. Si todo sale bien, un servidor DHCP en la red le responderá emitiendo una dirección y otra información necesaria para completar la configuración de red del cliente.

La asignación dinámica de direcciones es útil para configurar máquinas móviles o temporales. Las personas que se trasladan de oficina a oficina pueden conectar sus máquinas en la red local y obtener una dirección apropiada para su ubicación.

En este capítulo cubriremos el procedimiento para configurar un servidor DHCP y un cliente de DHCP. Esto incluye obtener e instalar el software necesario y luego completar el proceso de escribir su respectivo archivo de configuración. Al final del capítulo irá paso a paso por la puesta en marcha de un completo sistema de muestra.

NOTA DHCP es un estándar. De manera que, cualquier sistema operativo que pueda comunicarse con otros servidores y clientes de DHCP, podrá trabajar con las herramientas DHCP. Una solución común incluye el uso de un servidor DHCP basado en Linux en ambientes de oficina donde hay un gran número de clientes basados en Windows. Los sistemas Windows pueden configurarse para utilizar DHCP y contactar al servidor Linux para obtener sus direcciones IP. Los clientes de Windows no tienen por qué saber ni les importará que su configuración IP se las proporcione un servidor Linux, porque DHCP es un protocolo basado en estándares y la mayoría de las implementaciones tratan de apegarse a éste.

LA MECÁNICA DE DHCP

Cuando un cliente es configurado para obtener su dirección de la red, solicita una dirección con el formato de una solicitud DHCP. Un servidor DHCP escucha las solicitudes de los clientes. Cuando se recibe una solicitud, revisa su base de datos local y emite la respuesta apropiada. La respuesta siempre incluye la dirección y puede incluir servidores de nombres, máscaras de red y una puerta de enlace predeterminada (gateway). El cliente recibe la respuesta del servidor y configura sus parámetros locales con los datos recibidos.

El servidor DHCP mantiene una lista de direcciones que puede emitir. Cada dirección se emite por un *periodo de préstamo* durante el cual el cliente tiene autorización para utilizar la dirección asignada, antes de que sea momento de contactar otra vez al servidor para su renovación. Al término de dicho periodo, es de esperarse que el cliente ya no estará utilizando la dirección. En consecuencia, el servidor DHCP supone que la dirección vuelve a estar disponible y la regresa a su acervo de direcciones.

La implementación del servidor DHCP de Linux incluye varias características clave, comunes a varias implementaciones de servidores DHCP. Se puede configurar el servidor para que emita cualquier dirección libre en su acervo o para que emita direcciones específicas a una máquina determinada. Además de atender solicitudes DHCP, el servidor DHCP también atiende solicitudes BOOTP.

EL SERVIDOR DHCP

DHCPD, el servidor DHCP, es responsable de proporcionar direcciones IP y otra información relevante a solicitud de los clientes. Dado que el protocolo DHCP se basa en transmisión de señales, habrá de contar con la presencia de un servidor en cada subred donde se requiera proveer el servicio DHCP.

Instalación del software DHCP mediante RPM

El servidor ISC DHCP es la implementación de facto para máquinas Linux. En muchas distribuciones esta versión está disponible en formato RPM empaquetado.

En esta sección veremos el proceso de instalación del software ISC DHCP utilizando RPM. En sistemas Linux que utilizan Fedora Core o RHEL, el software ISC DHCP está separado en dos diferentes paquetes. Éstos son:

- ▼ **dhclient*.rpm** Este paquete proporciona el demonio del cliente de ISC DHCP.
- ▲ **dhcp*.rpm** Este paquete incluye los servicios y el agente de retransmisiones del servidor ISC DHCP.

En la mayoría de las distribuciones de Linux, lo más seguro es que ya tenga instalado el software DHCP que va del lado del cliente. Veamos si nuestro sistema muestra ya lo tiene instalado. Escriba

```
[root@servidorA ~]# rpm -qa | grep dhc  
dhcpv6_client-0.10-13  
dhclient-3.0.2-12
```

En la tercera línea de esta salida verá que el paquete dhclient ya está instalado.

Para poner en marcha el servidor DHCP necesitamos instalar el paquete correspondiente. Utilizaremos **up2date** para descargar e instalar el software en forma automática. Escriba

```
[root@servidorA ~]# up2date -i dhcp
```

Una vez que este comando sea completado con éxito, deberá tener instalado el software necesario.

Configuración del servidor DHCP

El archivo de configuración principal predeterminado del servidor ISC DHCP es **/etc/dhcpd.conf**. Este archivo encapsula dos distintos grupos de configuraciones:

- ▼ Un conjunto de declaraciones para describir redes, anfitriones o grupos conectados al sistema y tal vez un rango de direcciones que se pueden asignar a cada entidad. Se pueden usar varias declaraciones para describir diversos grupos de clientes. Las declaraciones pueden estar anidadas una dentro de otra cuando son necesarios varios conceptos para describir un conjunto de clientes o anfitriones.
- ▲ Un conjunto de parámetros que describe el comportamiento general de un servidor. Los parámetros pueden ser globales, o locales para un conjunto de declaraciones.

Descarga, compilación e instalación del software ISC DHCP desde el código fuente

Si el software ISC DHCP no está disponible en forma empaquetada en la distribución de Linux a su alcance, recuerde que siempre podrá construirlo a partir del código fuente disponible en el sitio de ISC en <http://www.isc.org>. También es posible que sólo quiera asegurarse de que tiene instaladas las más recientes enmiendas disponibles para el software, las cuales quizás no estén implementadas en su distribución.

Al momento de escribir esta obra, la versión más estable del software es la versión 3.0.3b3, misma que puede descargar directamente desde <ftp://ftp.isc.org/isc/dhcp/dhcp-3.0-history/dhcp-3.0.3b3.tar.gz>.

Una vez que el paquete ha sido descargado, desempaque el archivo como se muestra. Para este ejemplo supondremos que el código fuente se ha descargado en el directorio **/usr/local/src/**. Desempaque el archivo tar así:

```
[root@servidorA src]# tar xvzf dhcp-3.0.3b3.tar.gz
```

Entre al subdirectorio **dhcp*** creado por este comando y dedique un minuto para leer cualesquier archivos Readme presentes.

Enseguida configure el paquete con el comando **configure**.

```
[root@servidorA dhcp-3.0.3b3]# ./configure
```

Para compilar e instalar, ejecute los comandos **make**; **make install**

```
[root@servidorA dhcp-3.0.3b3]# make ; make install
```

La versión del software ISC DHCP construida desde el código fuente instala el demonio del servidor DHCP (**dhcpd**) dentro del directorio **/usr/sbin/** y el cliente de DHCP (**dhcpclient**) dentro del directorio **/sbin/**.

NOTA Como cada sitio tiene una red particular con direcciones particulares, es necesario que cada sitio sea puesto en marcha con su propio archivo de configuraciones. Si esta es la primera vez que lidia con DHCP, quizás quiera empezar con el archivo de configuración de muestra que presentamos hacia el final de este capítulo para modificarlo a fin de que refleje las características de su red.

Como sucede con la mayoría de los archivos de configuración en UNIX, el archivo está en texto ASCII y puede ser modificado con su editor de textos favorito. La estructura general del archivo es como sigue:

```
Parámetros globales;  
Declaración1  
[parámetros relacionados a declaración1]  
[subdeclaración anidada]
```

```
Declaración2
  [parámetros relacionados a declaración2]
  [subdeclaración anidada]
```

Como este esquema lo muestra, un bloque de declaraciones agrupa un conjunto de clientes. Puede aplicar parámetros distintos a cada bloque de declaraciones.

Declaraciones

Quizá quiera agrupar clientes diferentes por razones distintas, como requisitos organizacionales, distribución de la red y dominios administrativos. Para ayudar con la agrupación de estos clientes, introducimos las siguientes declaraciones.

group El listado individual de parámetros y declaraciones para cada anfitrión, repetido una y otra vez, puede hacer que el archivo de configuración sea difícil de manejar. La declaración **group** le permite aplicar un conjunto de parámetros y declaraciones a una lista de clientes, redes compartidas o subredes. La sintaxis para la declaración **group** es como sigue:

```
group etiqueta
  [parámetros]
  [subdeclaraciones]
```

donde **etiqueta** es el nombre definido por el usuario para identificar al grupo. El bloque **parámetros** contiene una lista de parámetros que se pueden aplicar al grupo. Las **subdeclaraciones** se usan en caso de que sea necesario un mayor grado de precisión para describir clientes adicionales que sean miembros de la declaración actual.

Por el momento ignore el campo de parámetros. Trataremos con más detalle dicho campo en la sección “Parámetros”, más adelante.

host Esta declaración se usa para aplicar un conjunto de parámetros y declaraciones a un anfitrión en particular además de los parámetros especificados para el grupo. Es común que ello se utilice para direcciones fijas al inicio o para los clientes de BOOTP. La sintaxis para una declaración **host** es como sigue:

```
host etiqueta
  [parámetros]
  [subdeclaraciones]
```

La **etiqueta** es el nombre definido por el usuario para identificar al anfitrión de grupo. Los **parámetros** y las **subdeclaraciones** ya se describieron en la declaración **group**.

shared-network Esta declaración agrupa un conjunto de direcciones de miembros dentro de la misma red física. Esto permite que los parámetros y las declaraciones se agrupen según diversos propósitos administrativos. La sintaxis es

```
shared-network etiqueta
  [parámetros]
  [subdeclaraciones]
```

La **etiqueta** es el nombre definido por el usuario para identificar a la red compartida. Los **parámetros** y las **subdeclaraciones** ya se describieron en la declaración previa.

subnet Esta declaración se usa para aplicar un conjunto de parámetros y/o declaraciones a un conjunto de direcciones que coinciden con la descripción de esta declaración. La sintaxis es como sigue:

```
subnet númeroDeSubred netmask máscaraDeRed
    [parámetros]
    [subdeclaraciones]
```

El **númeroDeSubred** es el red que quiere declarar como la fuente de direcciones IP que se asignarán a anfitriones individuales. La **máscaraDeRed** es eso mismo (vea el capítulo 12 para obtener más detalles sobre las máscaras de red) para una subred. Los **parámetros** y las **subdeclaraciones** ya se describieron en una declaración previa.

range Para el inicio dinámico, la declaración **range** especifica el rango de direcciones que serán válidas para ser asignadas a los clientes. La sintaxis es como sigue:

```
range [dynamic-bootp] direcciónInicial [direcciónFinal] ;
```

La palabra reservada **dynamic-bootp** se utiliza para alertar al servidor que el siguiente rango de direcciones es para el protocolo BOOTP. La **direcciónInicial** y la **direcciónFinal** opcional son las direcciones reales para el inicio y el final del bloque de direcciones IP. Se supone que los bloques son consecutivos y están en la misma subred de direcciones.

Parámetros

Hemos presentado este concepto en forma breve en páginas anteriores de este capítulo. La activación de estos parámetros afectará el comportamiento del servidor para un grupo relevante de clientes. Veremos estos parámetros en esta sección.

always-reply-rfc1048 La sintaxis de este parámetro es como sigue:

```
always-reply-rfc1048;
```

Se usa primordialmente para clientes de BOOTP. Hay clientes de BOOTP que requieren la respuesta de un servidor que sea clase BOOTP RFC 1048 al 100%. Si activa este parámetro asegura que este requisito sea cumplido.

authoritative La sintaxis de este parámetro es como sigue:

```
authoritative;
not authoritative;
```

Bajo condiciones normales el servidor DHCP supondrá que no sabe si es correcta y tiene la información de configuración sobre un segmento de red dado. Ello es así porque si un usuario instala accidentalmente un servidor DHCP sin comprender por completo cómo configurarlo, éste no enviará mensajes DHCPNAK espurios a clientes que han obtenido direcciones de un servidor DHCP legítimo dentro de la red.

default-lease-time La sintaxis de este parámetro es como sigue:

```
default-lease-time segundos;
```

El valor en **segundos** es la duración del periodo de préstamo que se asigna a una dirección IP si el cliente no especifica una duración específica.

dynamic-bootp-lease-cutoff La sintaxis de este parámetro es como sigue:

```
dynamic-bootp-lease-cutoff fecha;
```

Los clientes de BOOTP no saben acerca del concepto de periodo de préstamo. En forma predeterminada, el servidor DHCP asigna a los clientes de BOOTP direcciones IP que no caducan. Sin embargo, bajo ciertas condiciones sería más útil hacer que el servidor no asigne direcciones a un conjunto de clientes de BOOTP. En esos casos se utiliza este parámetro.

La **fecha** se especifica con el formato *W YYYY/MM/DD HH:MM:SS*, donde *W* es el día de la semana en formato **cron** (0=Domingo, 6=Sábado), *YYYY* es el año en formato de cuatro dígitos, *MM* es el mes (01=Enero, 12=Diciembre), *DD* es el día en dos dígitos, *HH* es la hora en dos dígitos y formato de 24 horas (00=Medianoche, 23=11 P.M.), *MM* es la representación de dos dígitos de los minutos y *SS* es la representación de dos dígitos de los segundos.

dynamic-bootp-lease-length La sintaxis de este parámetro es como sigue:

```
dynamic-bootp-lease-length segundos;
```

Aunque los clientes de BOOTP no tienen un mecanismo para que caduquen las direcciones IP que reciben, en ocasiones puede resultar seguro que el servidor suponga que la dirección asignada ya no está en uso a fin de liberarla para uso posterior. Ello es útil si sabe que la aplicación BOOTP dura poco tiempo. Si es así, el servidor puede fijarla en una cantidad de **segundos** a fin de que expire una vez transcurrido ese tiempo.

CUIDADO Esta opción debe utilizarse con precaución debido a que puede ocasionar problemas si emite una dirección antes de que otra haya suspendido su uso.

filename La sintaxis de este parámetro es como sigue:

```
filename nombreDelArchivo;
```

En algunas aplicaciones el cliente de DHCP quizás necesite saber el nombre del archivo utilizado para iniciar. Con frecuencia este parámetro se usa con **next-server** a fin de obtener un archivo remoto para configuración de instalación y arranque sin disco.

fixed-address La sintaxis de este parámetro es como sigue:

```
fixed-address dirección [, dirección ·];
```

Este parámetro sólo aparece en la declaración **host**. Especifica el conjunto de direcciones assignable a un cliente.

get-lease-hostnames La sintaxis de este parámetro es como sigue:

```
get-lease-hostnames [true | false];
```

Si este parámetro se establece en true (verdadero), el servidor buscará la representación de todas las direcciones dentro del alcance de la declaración y utilizará el resultado para la opción **hostname**.

hardware La sintaxis de este parámetro es como sigue:

```
hardware tipoDeHardware direcciónMACdelHardware;
```

A fin de que un cliente de BOOTP sea reconocido, la dirección de red del hardware debe ser declarada utilizando una cláusula **hardware** en instrucción **host**. Aquí el **tipoDeHardware** debe ser el nombre del tipo de interfaz física del hardware. Actualmente sólo se reconocen los tipos **ethernet** y **token-ring**.

La **direcciónMACdelHardware** (también conocida como dirección MAC) es la dirección física de la interfaz, representada con frecuencia como un conjunto de octetos hexadecimales delimitados por dos puntos (:). La instrucción **hardware** también se puede usar para clientes de DHCP.

max-lease-time La sintaxis de este parámetro es como sigue:

```
max-lease-time segundos;
```

Un cliente tiene la opción de solicitar la duración de un periodo de préstamo. La solicitud se aprueba en tanto el periodo de préstamo solicitado no exceda los **segundos** especificados en esta opción. De lo contrario, se otorga el número máximo de **segundos** especificado aquí.

next-server La sintaxis de este parámetro es como sigue:

```
next-server nombreDelServidor;
```

Esta instrucción especifica la dirección del anfitrión servidor desde el cual será cargado el archivo de arranque inicial (especificado en el statement **filename**). Aquí **nombreDelServidor** es una dirección IP numérica o un nombre de dominio.

server-identifier La sintaxis de este parámetro es como sigue:

```
server-identifier nombreDelAnfitrión;
```

Parte de la respuesta DHCP es la dirección para el servidor. En sistemas con varias interfaces el servidor DHCP asigna la dirección IP de la primera interfaz. Pero no siempre esa interfaz la alcanzan todos los clientes de un servidor o el alcance de una declaración. En esas raras situaciones este parámetro se puede usar para enviar la dirección IP de la interfaz correcta con la que el cliente debe comunicarse con el servidor. El valor especificado debe ser una dirección IP para el servidor DHCP y debe ser alcanzable por todos los clientes atendidos en ese campo de aplicación en particular.

server-name La sintaxis de este parámetro es como sigue:

```
server-name nombre;
```

Esta instrucción puede utilizarse para informar al cliente acerca del **nombre** del servidor desde el cual está iniciando. El **nombre** debe ser aquel que será proporcionado al cliente. Este parámetro suele ser útil para clientes remotos o aplicaciones de instalación de redes.

use-lease-addr-for-default-route La sintaxis de este parámetro es como sigue:

```
use-lease-addr-for-default-route [true|false];
```

Algunas configuraciones de red utilizan una técnica llamada *ProxyARP* de manera que un anfitrión pueda llevar control de otros anfitriones que están ubicados fuera de la subred. Si su red está configurada para ofrecer soporte a ProxyARP, tal vez quiera configurar al cliente para que se utilice a sí mismo como la ruta predeterminada. Ello forzará el uso del Address Resolution Protocol (ARP) (Protocolo de representación de direcciones) a fin de encontrar todas las direcciones remotas (fuera de la subred).

CUIDADO El comando **use-lease-addr-for-default-route** debe utilizarse con precaución, pues no cualquier cliente puede utilizar su propia interfaz como una ruta predeterminada.

Opciones

Actualmente el servidor DHCP ofrece soporte para más de 60 opciones. La sintaxis general de una opción es la siguiente:

```
option nombreDeLaOpción [modificadores]
```

En la tabla 27-1 se resumen las opciones DHCP utilizadas con mayor frecuencia.

Un archivo dhcpd.conf de muestra

El siguiente es un ejemplo de un sencillo archivo de configuración DHCP.

```
subnet 192.168.1.0 netmask 255.255.255.0
    # Opciones
    option routers 192.168.1.1;
    option subnet-mask 255.255.255.0;

    option domain-name "ejemplo.org";
    option domain-name-servers ns1.ejemplo.org;

    # Parámetros
    default-lease-time 21600;
    max-lease-time 43200;
    # Declaraciones
    range dynamic-bootp 192.168.1.25 192.168.1.49;

    # Declaraciones anidadas
    host clienteA
        hardware ethernet 00:80:c6:f6:72:00;
        fixed-address 192.168.1.50;
```

Opción	Descripción
broadcast-address	Una dirección en la subred del cliente especificada como la dirección de transmisión de señal
domain-name	El nombre de dominio que el cliente debe utilizar como el nombre de dominio local cuando busque anfitriones
domain-name-servers	La lista de servidores DNS que el cliente utilizará para representar nombres de anfitriones
host-name	La cadena de texto utilizada para identificar el nombre del cliente
nis-domain	El nombre del dominio NIS (Sun Network Information Services) del cliente
nis-servers	Una lista de servidores NIS disponible para el cliente
routers	Una lista de direcciones IP para routers que el cliente debe utilizar, en orden de preferencia
subnet-mask	La máscara de red que el cliente deberá utilizar

Tabla 27-1. Las opciones más comunes de dhcpd.conf

En este ejemplo se ha definido una sola subred. Se ha establecido que los clientes de DHCP utilicen 192.168.1.1 como su router predeterminado (dirección de la puerta de enlace) y 255.255.255.0 como la máscara de subred.

La información DNS se pasa a los clientes; utilizarán ejemplo.org como nombre de dominio y ns1.ejemplo.org como su servidor DNS.

Se establece un periodo de préstamo de 21 600, pero si los clientes solicitan periodos más largos, quizás les sean autorizados tiempos que pueden durar tanto como 43 200 segundos.

El rango de direcciones IP que serán emitidas empiezan en 192.168.1.25 y pueden ir tan alto como 192.168.1.49. La máquina con una dirección MAC de 00:80:c6:f6:72:00 siempre recibirá una dirección IP de 192.168.1.50.

Comportamiento general durante el tiempo de ejecución

Una vez que inicia, el demonio espera con paciencia a que llegue la solicitud de un cliente antes de que la procese. Cuando llega una solicitud, ésta se procesa y se emite una dirección; el demonio entonces registra la dirección en un archivo llamado **dhcpd.leases**. En sistemas Fedora y RHEL, este archivo se almacena en el directorio **/var/lib/dhcp/**.

EL DEMONIO DEL CLIENTE DE DHCP

El demonio del cliente de ISC DHCP (llamado **dhclient**), incluido en muchas de las distribuciones populares de Linux, es el componente de software utilizado para comunicarse con el servidor DHCP que se describió en secciones previas. Si se carga en memoria, intentará obtener una dirección desde un servidor DHCP y luego configurará sus propiedades de red en consecuencia.

Configuración del cliente de DHCP

Lo usual es iniciar el cliente desde los archivos de arranque, aunque también puede iniciararlo a mano. Debe iniciarse antes que cualquier otro servicio basado en red. Ello se debe a que cualesquier otros servicios de red son inservibles a menos que el sistema en sí sea capaz de acceder a la red.

Por otro lado, el cliente puede iniciarse desde una línea de comandos, en cualquier momento después del arranque. El demonio cliente puede iniciarse sin opciones adicionales... pero intentará obtener una dirección y un periodo de préstamo para todas las interfaces configuradas en el sistema.

Enseguida mostramos cómo iniciar el cliente desde una línea de comandos en su forma más básica:

```
[root@servidorB ~]# dhclient
.....<SALIDA TRUNCADA PARA ABREVIAR EL EJEMPLO>.....
Sending on   LPF/eth0/00:0c:29:f8:b8:88
Sending on   Socket/fallback
DHCPDISCOVER on lo to 255.255.255.255 port 67 interval 7
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 192.168.1.1
SIOCADDRT: File exists
bound to 192.168.1.36 -- renewal in 188238 seconds.
```

NOTA En sistemas Fedora Core o RHEL tiene a su disposición scripts de configuración de red que automáticamente ponen en marcha el sistema como un cliente de DHCP cada vez que reinicia, de manera que no tenga que ejecutar a mano el demonio **dhclient** cada vez que el sistema necesite una dirección IP. Para configurarlo, todo lo que necesita hacer es editar el archivo **/etc/sysconfig/network-scripts/ifcfg-eth*** asegurándose de que, cuando menos, la variable **BOOTPROTO** se establece a **dhcp** como se muestra en el siguiente listado:

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

De igual forma, puede iniciar el demonio cliente con banderas adicionales que modifiquen un poco el comportamiento del software. Por ejemplo, tiene la opción de especificar la interfaz (como eth0) para la cual debe solicitarse una dirección con periodo de préstamo.

La sintaxis completa del comando es la siguiente:

```
Usage: dhclient [-1dqr] [-nw] [-p <punto>] [-s servidor]
                [-cf archivo-config] [-lf archivo-préstamo] [-pf archivo-pid] [-e VAR=val]
                [-sf archivo-script] [interfaz]
```

Algunas de las opciones se describen en la tabla 27-2.

Opción	Descripción
-p	Especifica un puerto UDP distinto a ser utilizado por el cliente de DHCP, en vez del puerto estándar 68.
-d	Obliga al cliente de DHCP a que funcione como un proceso de primer plano, en vez de asumir su comportamiento normal como un proceso de segundo plano. Es útil para depuración.
-q	Esta bandera evita que cualesquiera mensajes que no sean errores sean impresos en el descriptor estándar de errores.
-r	Le dice en forma explícita al programa dhclient que libere la dirección asignada y, una vez que libera su periodo de préstamo, el cliente termina su ejecución.
-1	La bandera -1 ocasiona que dhclient intente sólo una vez para conseguir una dirección IP en periodo de préstamo. Si falla, termina su ejecución con un código de salida de dos.
-cf	Especifica la ubicación del archivo de configuración para dhclient . La ubicación predeterminada es /etc/dhclient.conf .
-lf	Especifica la ubicación de la base de datos del periodo de préstamos. El valor predeterminado es el archivo /var/lib/dhcp/dhclient.leases .
-pf	Define el archivo que guarda la ID de proceso de dhclient .
interfaz	Especifica la interfaz que dhclient habrá de configurar.

Tabla 27-2. Las opciones de dhclient en la línea de comandos

RESUMEN

DHCP es una herramienta útil para configurar en forma dinámica las direcciones para un grupo numeroso de máquinas o estaciones de trabajo móviles. Gracias a que DHCP es un protocolo abierto, la arquitectura y la plataforma tanto del servidor como del cliente son irrelevantes.

Una computadora que ejecuta Linux puede atender solicitudes DHCP. El software para configurar esto es altamente configurable y tiene mecanismos para persistir aun después de que se presenten fallas en las máquinas.

También existe software para configurar las propiedades de red de una máquina Linux desde un servidor DHCP en una red. El demonio cliente tiene una cantidad de opciones que le permiten hablar con una variedad de servidores DHCP.

CAPÍTULO 28



Copias
de seguridad

Un servidor que no se respalda es un desastre que está a punto de ocurrir. Realizar copias de seguridad es una parte crítica del mantenimiento de cualquier servidor, sin importar el sistema operativo que tenga. En este capítulo, veremos las opciones que Linux trae de fábrica para generar copias de seguridad (o respaldos). También existen varios paquetes comerciales que puede adquirir cuyos precios oscilan entre unos cuantos cientos hasta varios miles de dólares. El mejor paquete para usted dependerá de su sitio y de sus necesidades.

EVALUACIÓN DE NECESIDADES PARA COPIAS DE SEGURIDAD

La implementación de una solución para copias de seguridad no es una tarea trivial. Es preciso que considere las interacciones entre las partes de su red, los servidores y los recursos distribuidos entre éstos. Aún más delicado es decidir el orden en el que se elaborarán las copias de seguridad. Por ejemplo, si quiere respaldar varias particiones en paralelo, ¡perderá los beneficios de ese paralelismo si hay algún conflicto en el bus SCSI! Y, desde luego, debe hacer los arreglos necesarios para que las copias de seguridad sean generadas y verificadas con cierta regularidad.

Por desgracia, no existe un libro de recetas para poner en marcha la elaboración de copias de seguridad en una red. Cada organización tiene sus propias necesidades basadas en su(s) sitio(s), su red, y su propio patrón de crecimiento. Para tomar una decisión informada necesita considerar preguntas como las siguientes:

- ▼ ¿Cuántos datos necesita respaldar?
- ¿Qué tipo de hardware utilizará para llevar a cabo la elaboración de copias de seguridad?
- ¿Qué tanto ancho de banda necesitará en su red para realizar los respaldos?
- ¿Qué tan rápido necesitará restaurar los datos respaldados?
- ¿Qué tan seguido tendrá que restaurar las copias de seguridad?
- ▲ ¿Qué tipo de administración de grabación necesita?

¿Cuántos datos?

El aspecto más importante para definir las necesidades de respaldos en su red es determinar con exactitud cuántos datos necesitará proteger mediante copias de seguridad. Esta pregunta es difícil de responder porque su determinación debe anticiparse al crecimiento. Considerando que casi siempre tendrá un presupuesto limitado, la planeación de copias de seguridad deberá hacerse con la mayor anticipación financiera posible.

También es importante considerar la frecuencia con la que cambian sus datos. Los datos que cambian con regularidad (como las bases de datos) necesitan respaldarse con frecuencia y celeridad, mientras que los datos que casi no cambian (como el directorio /etc) no necesitan respaldarse con la misma regularidad (y quizás ni siquiera lo necesiten).

Cuando examine sus requerimientos sea cauteloso y considere los datos comprimibles de los no comprimibles. Conforme los discos duros locales aumentan en capacidad, los individuos los ocupan para almacenar inmensas colecciones personales de música e imágenes que no tienen nada que hacer en su organización (en el caso de las copias de seguridad personales aplica el criterio opuesto). Lo más prudente será dar a conocer por escrito las políticas de su organización sobre el

tema, de manera que las expectativas sean claras: si los usuarios creen que usted respalda todo el contenido de sus sistemas, se llevarán una desagradable sorpresa cuando vean que su música y sus imágenes personales no estaban incluidas en las copias de seguridad. En el otro lado de la moneda usted quizás tenga que enfrentar de improviso un repentino incremento en las necesidades de almacenamiento cuando un usuario descubra P2P o traiga consigo su colección de MP3 a trabajar.

¿Qué tipo de medios de almacenamiento?

El tipo de hardware que elija debe permitirle almacenar la cantidad de datos que requiere respaldar, la frecuencia con la que debe respaldarlos y, de ser necesario, facilitarle el traslado de las copias a sitios de almacenamiento fuera de sus instalaciones.

Existen cuatro opciones comunes: cintas magnéticas, discos duros, CD y DVD grabables. De éstas, las cintas son las veteranas y además ofrecen la más amplia gama de densidades, formas y mecanismos.

Entre sus opciones de cintas puede ser difícil seleccionar el tipo específico. Muchas de las opciones de alta densidad son atractivas por una razón obvia: puede amontonar más datos en una sola cinta. Claro, es típico que las cintas de alta densidad y las unidades de las cintas cuesten más. Es obvio que deberá buscar aquella solución óptima que le permita almacenar la mayor cantidad de datos al mejor precio que esté en equilibrio con sus necesidades.

NOTA Muchos anuncios de unidades de cintas alardean impresionantes capacidades de almacenamiento. Tenga presente que estas cifras corresponden a datos comprimidos en la cinta, no a su capacidad real. Es usual que la cantidad de datos sin comprimir sea alrededor de la mitad de la capacidad para datos comprimidos. Es importante resaltarlo porque los algoritmos de compresión alcanzan distintos niveles de compresión dependiendo del tipo de datos que se esté almacenando. Por ejemplo, los archivos de texto se comprimen bastante bien. Algunos formatos de gráficos y sonido no se comprimen nada. Cuando estime la cantidad de datos que tendrá que respaldar en una sola unidad asegúrese de considerar la mezcla de datos en sus servidores.

Los sistemas para elaboración de copias de seguridad que se basan en discos duros son un fenómeno relativamente nuevo. El concepto es sencillo: si el propósito principal de una copia es proteger contra accidentes simples (eliminación de archivos, fallas físicas en discos, etc.), entonces esta tecnología le funciona. Las transferencias son rápidas, los medios (discos) son baratos y, por menos de 1 000 dólares, podrá construir un sistema RAID casero utilizando una PC de bajo costo, una controladora RAID barata y unos cuantos discos duros comunes (las soluciones comerciales, como aquellas de NetApp, ofrecen más opciones para necesidades de mayor capacidad). Si utiliza este método podrá automatizar la elaboración de copias de seguridad sin necesidad de cambiar cintas y, además, podrá añadir capacidad adicional a bajo costo. La desventaja de este método es que el almacenamiento en otro sitio no es posible a menos que implemente una solución combinada con medios de almacenamiento removibles. Aun cuando es posible encontrar soluciones en las que puede reemplazar o añadir discos “en caliente”, no son tan robustas como las cintas si compara la facilidad de su manejo y traslado (una cinta puede caerse varias veces al suelo sin mayor riesgo. ¡No puede decirse lo mismo de un disco duro!)

Con los CD y DVD grabables, las copias de seguridad de bajo costo se han hecho posibles. Estos medios de almacenamiento son fáciles de utilizar y son muy baratos. Las desventajas son que su capacidad de almacenamiento está muy por debajo de sus contrapartes y que su tiempo de vida es cuestionable y deja mucho que desechar. Si la cantidad de datos que necesita respaldar no es muy grande o los datos no cambian una vez grabados (como las fotografías que toma con

una cámara digital), este tipo de medios funciona bien. El tiempo de vida del medio varía según el fabricante. Para sus copias de seguridad no olvide que debe asegurar algunos años de vida.

Está creciendo la popularidad de combinar medios fijos y medios removibles. En soluciones combinadas, los respaldos regulares se hacen sobre disco duro y, de vez en cuando, del disco se pasa a cintas magnéticas.

Como sea, haga planes para el día en que fallen sus medios de almacenamiento. Es decir, haga planes para mover sus datos respaldados a nuevos medios después de que transcurran algunos años. Ello es necesario no sólo para asegurarse de que está utilizando medios recién adquiridos, sino para que la unidad tenga oportunidad de "ejercitarse" y no pierda la calibración de modo que un equipo más moderno aun pueda leer y escribir en él. En algunos casos deberá considerar el formato en el que se graban los datos. Después de todo, a nadie le sirve un medio en buenas condiciones si no tiene manera de leer su contenido (es decir, considere si hoy le sería posible leer un disco flexible de su primera computadora).

Consideraciones sobre el desempeño de cintas magnéticas

Si se inclina por las copias de seguridad basadas en cintas magnéticas, considere dónde ubicará la unidad de la cinta y a qué sistema estará conectado. ¿Lo pondrá en un servidor que de por sí ya está atareado con muchos discos duros activos? ¿Lo ubicará en un servidor dedicado que cuente con varios discos para administrar la llegada de datos? ¿Hay limitaciones en el bus utilizado para transferir datos hacia el dispositivo? (por ejemplo, si está respaldando a una unidad de cinta SCSI, ¿tiene la cadena SCSI otros dispositivos que podrían estar ocupados?).

Por último, ¿es posible alimentar datos suficientemente rápido a la unidad de cinta para que los almacene? Si la unidad de cinta no recibe un flujo suficiente, detendrá la escritura hasta que obtenga más datos. La demora puede ser de varios segundos en un mecanismo lento, mientras la unidad realinea la cinta con el mecanismo y encuentra la primera posición disponible para escribir datos. Aun cuando la pausa sea breve, cuando ésta ocurre miles de veces durante una sola operación de respaldo, la acumulación incrementará en varias horas el tiempo de elaboración.

¿Cuánto ancho de banda necesitará en su red?

Es tristemente usual que el ancho de banda de la red se olvide en la planeación de las operaciones de respaldo. ¿Qué tantas ventajas puede obtener a partir de un servidor ultrarrápido una unidad de cintas magnéticas si les alimenta datos con una cerbatana?

Tome suficiente tiempo para entender la infraestructura de su red. Vea de dónde vienen los datos y a dónde van. Utilice herramientas SNMP, como MRTG (<http://www.mrtg.org>), para generar estadísticas acerca de sus comutadores y routers. Si necesita copias de seguridad de máquinas que están conectadas mediante concentradores sencillos (hubs), considere una secuencia de respaldos que no trabaje al mismo tiempo con dos máquinas dentro del mismo dominio de colisión.

Al reunir toda esta información podrá estimar el ancho de banda necesario para realizar copias de seguridad. Con su análisis le será posible encontrar qué actualizaciones a su red le permitirán obtener el mejor retorno de su inversión.

¿Qué velocidad de restauración?

Cuando recibe solicitudes para restaurar datos almacenados en cintas magnéticas seguramente estará bajo presión para entregar los datos de vuelta al usuario tan rápido como le sea posible. El tiempo que sus usuarios deberán esperar dependerá de la tecnología utilizada para la elaboración de respaldos.

Ello significa que deberá incorporar el costo relacionado con el tiempo de respuesta en la evaluación del sistema que implementará para sus copias de seguridad. Este punto puede resumirse en una pregunta: ¿cuánto está dispuesto a invertir para obtener el tiempo de respuesta que usted requiere?

Las restauraciones a partir de sistemas basados en discos duros son las más rápidas. Además, ofrecen la posibilidad de realizar respaldos en línea a través de un servidor de respaldos que los mismos usuarios pueden visitar para extraer ellos mismos los archivos que pretendan recuperar. Los CD y DVD grabables son relativamente igual de rápidos que los discos duros por las mismas razones. En comparación, las cintas magnéticas son mucho más lentas. Es necesario encontrar, leer y extraer un archivo individual en la cinta. Según la velocidad de la cinta y la ubicación del archivo (son sistemas de acceso lineal), ello puede tardar algún tiempo.

¿Cómo administrará las cintas?

Conforme crezca el volumen de sus copias de seguridad, así también se incrementará la necesidad de administrar los respaldos que hace. Éste es el punto donde algunas herramientas comerciales hacen su aparición. Cuando evalúe sus opciones, asegúrese de considerar la indexación y la administración de cintas. No le ofrece ninguna ventaja tener 50 cintas repletas de respaldos si no puede encontrar el archivo correcto. Por desgracia, este problema sólo crece conforme necesita más y más cintas para las copias de seguridad que elabora cada noche.

MANEJO DEL DISPOSITIVO DE CINTAS MAGNÉTICAS

El dispositivo de cinta interactúa con Linux como lo hace cualquier otro dispositivo: como un archivo. El nombre del archivo dependerá del tipo de dispositivo de cinta, el modo de operación elegido (autorrebobinado o sin rebobinado) y cuántas unidades estén conectadas al sistema.

Las unidades de cinta SCSI, por ejemplo, utilizan el siguiente esquema de denominación:

Nombre del dispositivo	Propósito
<code>/dev/stx</code>	Dispositivo de cinta SCSI de autorrebobinado; x es el número de la unidad de cinta. La numeración de las unidades depende del orden que éstas tengan en la cadena SCSI.
<code>/dev/nstx</code>	Dispositivo de cinta SCSI sin rebobinado; x es el número del dispositivo de cinta. La numeración de las unidades depende del orden que éstas tengan en la cadena SCSI.

NOTA Bajo devfs, el esquema de denominación ha cambiado a `/devfs/scsi/hostA/busB/targetT/lunL/mt` para las cintas con autorrebobinado y `/devfs/scsi/hostA/busB/targetT/lunL/mtn` para las cintas sin rebobinado. En ambos casos, A se refiere al número de tarjeta del anfitrión, B al número del canal, T a la ID del dispositivo, y L al dispositivo lógico en un dispositivo físico. (La designación del dispositivo lógico variará de un fabricante a otro. En el caso de los dispositivos de cinta con mecanismos múltiples, podría referirse al número del mecanismo.)

Digamos que tiene una sola unidad de cinta SCSI. Puede acceder a ella utilizando cualquiera de los siguientes nombres de archivo: `/dev/st0` o `/dev/nst0`. Si utiliza `/dev/st0`, la unidad rebobinará en forma automática la cinta después de que se escriba cada archivo. Por otro lado, si utiliza `/dev/nst0`, puede escribir un solo archivo a la cinta, marcar el final del archivo pero dejar la cinta en la posición donde se quedó. Ello le permite escribir múltiples archivos en una sola cinta.

NOTA Evidentemente, los dispositivos que no son SCSI usan una nomenclatura distinta. La desventaja es que no hay una manera estándar de referirse a estos dispositivos si no son SCSI. La controladora de cinta QIC-02, por ejemplo, utiliza la serie **/dev/tpqjc*** para los nombres de archivos. Si utiliza una unidad que no es SCSI, necesitará encontrar la documentación de los controladores correspondiente para saber qué nombre de dispositivo utilizará.

Quizá le sea útil definir un enlace simbólico (symlink) desde **/dev/tape** hacia el nombre correcto del dispositivo en modo de rebobinado; y otro symlink desde **/dev/nrtape** hacia el nombre de la unidad en modo sin rebobinado (por ejemplo, **/dev/tape → /dev/st0** y **/dev/nrtape → /dev/nst0**). Ello hará más sencillo recordar el nombre del dispositivo correcto cuando le envíe comandos. Vea el capítulo 5 para más información sobre el uso de **ln** para la creación de enlaces simbólicos.

Lo que hace diferentes a estos dispositivos de respaldo de los archivos en disco es que no cuentan con una estructura de sistema de archivos. Los archivos se escriben en forma continua en la cinta hasta que se llena o hasta que se escribe una marca de fin de archivo. Si una unidad de cinta está en modo sin rebobinado, la cabeza de escritura se queda en la posición inmediata después de la última marca de fin de archivo que fue escrita, lista para escribir el siguiente archivo.

Piense en los dispositivos de cinta en forma similar a un libro con capítulos. La encuadernación de páginas, así como el papel mismo, proveen un lugar donde puede poner palabras (los archivos). Son las marcas del editor (la aplicación para respaldos) las que separan todo el libro en secciones más pequeñas (archivos). Si usted (el lector) fuera una unidad de cinta de autorrebobinado, rebobinaría la cinta cada vez que terminara de leer una sección para después buscar desde el inicio la siguiente posición (capítulo) para seguir leyendo. Si, por el contrario, usted fuera una unidad de cinta sin rebobinado, dejaría el libro abierto en la última página que leyó para luego retomar desde ahí su lectura.

Uso de mknod y scsidev para crear los archivos del dispositivo

Si no tiene el archivo **/dev/st0** o el **/dev/nst0**, puede crear uno utilizando el comando **mknod** (vea el capítulo 5 para obtener una explicación de **mknod**). El número mayor para una unidad de cinta SCSI es 9, mientras que el número menor dicta de qué unidad se trata y si tiene autorrebobinado o no. Los números 0 a 15 representan las unidades 0 a 15, auto-rebobinado. Los números 128 a 143 representan las unidades 0 a 15, sin rebobinado. La unidad de cinta es un dispositivo de caracteres.

De tal manera que para crear **/dev/st0**, utilizaríamos el comando **mknod** como sigue:

```
[root@servidorA /root]# mknod /dev/st0 c 9 0
```

Y para crear **/dev/nst0**, utilizaríamos el comando **mknod** como sigue:

```
[root@servidorA /root]# mknod /dev/nst0 c 9 128
```

Otra opción para crear los nombres de los dispositivos es utilizar el programa **scsidev**. Esto creará entradas para los dispositivos dentro del directorio **/dev/scsi** que reflejarán el estado actual de su hardware SCSI con el tipo de dispositivo apropiado (bloque o carácter) y sus correspondientes números mayor y menor. La desventaja es que este método tiene aun otro esquema de nomenclatura.

La nomenclatura para dispositivos de cinta magnética creados utilizando scsidev es como sigue: **/dev/scsi/sthA-0cBiT1L**, donde **A** es el número del anfitrión, **B** es el número del canal, **T** es la ID blanca y **L** es el número de unidad lógica (lun, logical unit number).

Tantos esquemas de nomenclatura pueden parecerle frustrantes, lo cual es comprensible. Sin embargo, la clave en todos ellos es que de todas formas utilizan los mismos números mayor y menor. En otras palabras, ¡todos se refieren al mismo controlador! Al final podría decidir bautizar a sus dispositivos de cinta de autorrebobinado y sin rebobinado como “Sofía” y “Alessandra”, respectivamente, mientras ambas tengan los números mayores y menores correctos.

Manipulación de dispositivos de cinta magnética con mt

El programa **mt** proporciona controles simples para unidades de cinta, como rebobinar la cinta, expulsar la cinta o buscar un archivo en particular en la cinta. En el contexto de las copias de seguridad, **mt** es más útil como un mecanismo para rebobinar y buscar.

Todas las acciones de **mt** se especifican en comandos lanzados desde una línea de comandos. La tabla 28-1 muestra los parámetros para el comando.

Parámetro de comando mt	Descripción
-f dispositivoDeCinta	Especifica el <i>dispositivoDeCinta</i> . El primer dispositivo de cinta SCSI sin rebobinado es /dev/nst0 .
fsf número	Avanza la cinta el <i>número</i> de archivos especificado. La cinta se posiciona en el primer bloque del siguiente archivo; por ejemplo, fsf 1 dejaría la cabeza lectora lista para leer el segundo archivo de la cinta.
ASF posición	Ubica la cinta al inicio del archivo que se encuentra en la <i>posición</i> indicada por el parámetro. Ello se logra rebobinando la cinta y luego avanzando archivos hasta la <i>posición</i> indicada.
rewind	Rebobina la cinta.
erase	Borra la cinta.
status	Proporciona el estado de la cinta.
offline	Saca a la unidad de línea y, si es aplicable, expulsa la cinta.
load	Carga la cinta (sólo aplicable a ciertos dispositivos de cinta).
lock	Activa el seguro de la puerta del dispositivo (sólo aplicable a ciertas unidades de cinta).
unlock	Libera el seguro de la puerta del dispositivo (sólo aplicable a ciertas unidades de cinta).

Tabla 28-1. Parámetros del comando **mt**

NOTA Si no utiliza un dispositivo de cinta sin rebobinado, la unidad de cinta rebobinará en forma automática después de realizar alguna operación con `mt`. ¡Ello puede ser frustrante si está buscando un archivo específico!

- ▼ Para rebobinar la cinta en `/dev/nst0`, utilice el siguiente comando:

```
[root@servidorA /root]# mt -f /dev/nst0 rewind
```

- ▲ Para mover la cabeza de manera que esté lista para leer el tercer archivo en la cinta, utilice este comando:

```
[root@servidorA /root]# mt -f /dev/nst0 asf 2
```

HERRAMIENTAS DESDE LA LÍNEA DE COMANDOS

Linux viene con varias herramientas que le ayudan a generar sus copias de seguridad. Aunque carecen de interfaz administrativas frontales, su manejo es sencillo, y hacen el trabajo. Varios paquetes para respaldos formales en realidad utilizan estas herramientas como su mecanismo de respaldos base.

dump y **restore**

La herramienta `dump` trabaja haciendo una copia de un sistema de archivos completo. La herramienta `restore` puede tomar esta copia y extraer cualquier y todos los archivos de ella.

Para mantener copias de seguridad incrementales, `dump` utiliza el concepto de *niveles de vaciado*. Un nivel de vaciado 0 (cero) significa un respaldo completo. Cualquier nivel de vaciado superior a cero es un nivel incremental respecto de la última vez que ejecutó `dump` con un nivel menor. Por ejemplo, un nivel de vaciado 1 cubre todos los cambios en el sistema de archivos desde que ocurrió el último vaciado de nivel 0; un nivel de vaciado 2 cubre todos los cambios en el sistema de archivos desde que ocurrió el último vaciado de nivel 1, y así en adelante, hasta el nivel 9.

Considere un caso en el que tiene tres vaciados: el primero es de nivel 0, el segundo es de nivel 1 y el tercero también es de nivel 1. El primero es, desde luego, el respaldo completo. El segundo vaciado (nivel 1) contiene todos los cambios desde el primer vaciado. El tercer vaciado (también de nivel 1) *también* tiene todos los cambios desde el último nivel 0. Si generara un cuarto vaciado de nivel 2, éste contendría todos los cambios desde el tercer vaciado de nivel 1.

La herramienta `dump` guarda la información sobre sus vaciados en el archivo `/etc/dumpdates`. Este archivo lista cada sistema de archivos que se respaldó, cuándo se respaldó y a qué nivel. Con esta información es posible determinar cuál cinta utilizar para la restauración. Por ejemplo, si realiza vaciado de nivel 0 los lunes, incrementales de nivel 1 los martes y miércoles, y luego incrementales de nivel 2 los jueves y viernes, un archivo que se modificó por última vez el martes pero se borró accidentalmente el viernes puede ser restaurado del respaldo incremental del martes en la noche. Un archivo que se modificó durante la semana pasada estará en la cinta de nivel 0 del lunes.

NOTA La herramienta `dump` se envía con todas las distribuciones de Linux. Esta herramienta guarda una estrecha relación con el sistema de archivos. Por esta razón, la versión para Linux sólo funciona sobre sistemas de archivos nativos de Linux (ext2 y ext3). Si utiliza otro sistema de archivos como ReiserFS, JFS o XFS, asegúrese de utilizar la herramienta `dump` apropiada.

Uso de dump

La herramienta **dump** se utiliza desde una línea de comandos. Acepta muchos parámetros pero los más relevantes se muestran en la tabla 28-2.

Parámetro del comando dump	Descripción
-n	Define el nivel de vaciado, donde n es un número entre 0 y 9.
-a	Dimensiona la cinta en forma automática. Este es el comportamiento predeterminado de dump si ninguno de los parámetros -b , -B , -d o -s (como se documenta más adelante en esta misma tabla) se especifican.
-j	Utiliza la compresión bzip2. Note que bzip2, aunque ofrece un excelente esquema de compresión, requiere de mayor poder de cómputo (tiempo de CPU). Si utiliza este sistema de compresión, asegúrese de que su sistema es suficientemente rápido como para suministrar datos al dispositivo de cinta magnética sin ocasionar pausas. También tenga en cuenta que esta opción puede interrumpir la compatibilidad con otros sistemas UNIX.
-z	Utiliza la compresión gzip. Tenga en cuenta que esta opción puede interrumpir la compatibilidad con otros sistemas UNIX.
-b <i>número</i>	Ajusta el tamaño del bloque de vaciado al <i>número</i> especificado; se mide en kilobytes.
-B <i>número</i>	Especifica el <i>número</i> de registros por cinta que se vaciarán. Si hay más datos que vaciar que cinta disponible, dump le pedirá que inserte una nueva cinta.
-f <i>ubicación</i>	Especifica la <i>ubicación</i> del archivo resultante. Puede hacer que el archivo de vaciado sea un archivo normal que reside en otro sistema de archivos, o puede escribirlo en el dispositivo de cinta magnética. El dispositivo de cinta SCSI es /dev/st0 .
-u	Actualiza el archivo /etc/dumpdates después de un vaciado exitoso.
-d <i>densidad</i>	Especifica la <i>densidad</i> de la cinta en bits por pulgada.
-s <i>longitud</i>	Especifica la <i>longitud</i> de la cinta en pies.
-W	Muestra cuáles sistemas de archivos necesitan vaciarse pero sin realizar ningún vaciado. Esto se logra basándose en la información contenida en los archivos /etc/dumpdates y /etc/fstab .
-L <i>etiqueta</i>	Utiliza la <i>etiqueta</i> para nombrar el vaciado de manera que lo pueda leer el comando restore .
-S	Realiza una estimación del tamaño del vaciado sin realizarlo.

Tabla 28-2. Parámetros relevantes de la herramienta **dump**

Por ejemplo, este es el comando usado para llevar a cabo un vaciado de nivel 0 del sistema de archivos `/dev/hda1` en el dispositivo `/dev/st0`:

```
[root@servidorA /root]# dump -0 -f /dev/st0 /dev/hda1
```

Omisión del cálculo del tamaño de la cinta La herramienta `dump` requiere saber el tamaño de la cinta con la que está trabajando. Utiliza esta información para proporcionar respaldos multivolumen de manera que pueda pedir al operador que inserte la siguiente cinta cuando esté lista. Pero si no sabe el tamaño de su cinta y la opción `-a` no puede calcularlo, quizás pueda saber si el vaciado cabrá en la cinta (por ejemplo, quizás sepa que la partición que está copiando es de 2GB y la capacidad de la cinta, sin comprimir, es de 5GB). En esta situación puede realizar un truco para evitar que `dump` calcule el tamaño de la cinta. En vez de que vacíe directo al dispositivo, envíe la salida a la salida estándar y luego utilice el programa `cat` para redirigir el vaciado hacia la cinta. Retomando el ejemplo de la sección anterior, utilizaría el siguiente comando:

```
[root@servidorA /root]# dump -0 -f - /dev/hda1 | cat >> /dev/st0
```

Como está enviando la salida a la salida estándar, puede utilizar esta oportunidad para aplicar sus propios filtros de compresión al torrente de datos en vez de confiar en la compresión por hardware o la compresión interconstruida en los parámetros que proporciona a la línea de comandos. Por ejemplo, para utilizar `gzip` para comprimir su vaciado, escribiría

```
[root@servidorA /root]# dump -0 -f - /dev/hda1 | gzip --fast -c >> /dev/st0
```

CUIDADO Se considera peligroso vaciar sistemas de archivos que están siendo activamente utilizados. La única manera de estar 100% seguro de que un sistema no está en uso es desmontarlo primero. Por desgracia, muy pocas personas pueden darse el lujo de desmontar un sistema durante el tiempo necesario para respaldarlo. Lo mejor que puede hacer en estos casos es la inefable tarea de verificar las copias de seguridad con cierta regularidad. La verificación se hace mejor haciendo pruebas para ver si el programa `restore` (presentado en "Uso de `restore`" más adelante) puede leer por completo la cinta y extraer archivos de ésta. Es tedioso y no es divertido. Pero se sabe que muchos administradores de sistemas han perdido su trabajo debido a respaldos defectuosos, ¡no sea uno de ellos!

Uso de `dump` para respaldar todo un sistema La herramienta `dump` sólo trabaja haciendo un archivo de un solo sistema de archivos. Si su sistema está integrado por varios sistemas de archivos, necesitará ejecutar `dump` para cada uno de ellos. Debido a que `dump` crea su salida como un solo archivo enorme, puede almacenar múltiples vaciados en una sola cinta si utiliza un dispositivo sin rebobinado.

Suponiendo que va a respaldar en un dispositivo de cinta magnética SCSI, `/dev/nst0`, primero deberá decidir cuáles sistemas de archivos quiere respaldar. Encontrará esta información en el archivo `/etc/fstab`. Es obvio que no querrá respaldar archivos como `/dev/cdrom`, así que podrá omitirlos. Dependiendo de sus datos, querrá o no respaldar ciertas particiones (como `swap` y `/tmp`).

Asumamos que sólo quiere respaldar `/dev/hda1`, `/dev/hda3`, `/dev/hda5` y `/dev/hda6`. Para respaldar éstas a `/dev/nst0`, comprimiéndolas de paso, tendrá que emitir la siguiente serie de comandos:

```
[root@servidorA /root]# mt -f /dev/nst0 rewind
[root@servidorA /root]# dump -0uf - /dev/hda1 | gzip --fast -c >> /dev/nst0
[root@servidorA /root]# dump -0uf - /dev/hda3 | gzip --fast -c >> /dev/nst0
[root@servidorA /root]# dump -0uf - /dev/hda5 | gzip --fast -c >> /dev/nst0
[root@servidorA /root]# dump -0uf - /dev/hda6 | gzip --fast -c >> /dev/nst0
[root@servidorA /root]# mt -f /dev/nst0 rewind
[root@servidorA /root]# mt -f /dev/nst0 eject
```

El primer comando `mt` es para asegurarnos de que la cinta está completamente rebobinada y lista para aceptar datos. Luego vienen los comandos `dump` para cada una de las particiones, con sus salidas enfiladas a través de `gzip` antes de irse a la cinta. Para hacer los respaldos un poco más rápidos, la opción `--fast` se utiliza con `gzip`. Ello da por resultado una compresión que no es tan buena como el `gzip` normal, pero que es mucho más rápida y requiere menos tiempo de CPU. La opción `-c` en `gzip` le dice que envíe su salida a la salida estándar. Al final rebobinamos la cinta y la expulsamos.

Uso de restore

El programa `restore` lee los archivos de vaciado creados por `dump` y extrae archivos individuales y directorios de éstos. Aunque `restore` es una herramienta para la línea de comandos, ofrece un modo más intuitivo e interactivo que le permite ir a través de la estructura de un directorio a partir de la cinta.

En la tabla 28-3 se muestran las opciones utilizadas con esta herramienta en la línea de comandos.

Una restauración típica Una solicitud típica a `restore` es como sigue:

```
[root@servidorA /root]# restore -ivf /dev/st0
```

Esto extraerá el archivo de vaciado del dispositivo en `/dev/st0` (el primer dispositivo de cinta SCSI), imprimirá el resultado de cada paso de `restore`, y luego proveerá una sesión interactiva con la cual podrá decidir cuáles archivos habrá de restaurar del vaciado.

Una restauración completa En caso de que un sistema de archivos se pierda por completo, puede recrear el sistema de archivos utilizando el comando `mke2fs` y luego `restore` para poblar el sistema de archivos. Por ejemplo, digamos que falla nuestra unidad SCSI externa (`/dev/sda`), el cual tiene una sola partición (`/dev/sda1`). Después de reemplazarla por una unidad nueva, volveríamos a crear el sistema de archivos como sigue:

```
[root@servidorA /root]# mke2fs /dev/sda1
```

Enseguida, tenemos que montar la partición en la ubicación apropiada. Supongamos que ésta es la partición `/home`, así que escribiremos lo siguiente:

```
[root@servidorA /root]# mount /dev/sda1 /home
```

Opción herramienta	Descripción
restore	
-i	Habilita el modo interactivo de restore . La herramienta leerá el contenido del directorio de la cinta y luego presentará una interfaz tipo shell con la cual podrá navegar entre los directorios a fin de marcar los archivos que quiera restaurar. Una vez que haya marcado todos los archivos que quiera, restore recorrerá el vaciado en la cinta y restaurará esos archivos. Este modo es útil para restaurar archivos individuales, sobre todo si no está seguro del directorio en el que están.
-r	Reconstruye un sistema de archivos. En el caso de perderlo todo dentro de un sistema de archivos (una falla física del disco, por ejemplo), puede recrear un sistema de archivos vacío y luego restaurar todos los archivos y directorios desde el vaciado.
-b número	Establece el tamaño del bloque de vaciado a un número de kilobytes. Si no proporciona esta información, restore se las arreglará para obtenerla.
-f nombre	Lee el vaciado desde un archivo que se llame según el nombre especificado.
-T directorio	Especifica el directorio temporal para las actividades de restauración. El valor predeterminado es /tmp .
-v	Activa las instrucciones; explica cada paso que restore va realizando.
-y	En caso de presentarse un error, vuelve a intentar la operación en forma automática en vez de preguntarle al usuario si quiere reintentar.

Tabla 28-3. Opciones de la herramienta **restore** en la línea de comandos

Por último, con la cinta del vaciado en la unidad SCSI (**/dev/st0**), ejecutamos la restauración con el siguiente comando:

```
[root@servidorA /root]# cd /home; restore -rf /dev/st0
```

SUGERENCIA Si utilizó **gzip** para comprimir el vaciado, usted deberá descomprimirlo antes de que **restore** pueda procesarlo. Tan sólo pida a **gzip** que descomprima el dispositivo de cinta magnética y envíe la salida a la salida estándar. La salida estándar deberá ser enfilara hacia **restore** con la opción **-f** establecida para que la lea de la entrada estándar. He aquí el comando:

```
[root@servidorA /root]# gzip -d -c /dev/st0 | restore -ivf -
```

tar

En el capítulo 5 abordamos el uso de **tar** para crear *colecciones* de archivos. Lo que no vimos en ese momento fue que la intención original del programa **tar** es la de generar colecciones de archivos sobre cinta magnética (**tar** = **tape archive**). Debido a la naturaleza flexible de Linux para tratar dispositivos de la misma forma en que trata archivos, hemos estado utilizando **tar** como un medio para comprimir/descomprimir un grupo de archivos hacia/desde un solo archivo en el disco duro. Esos mismos comandos **tar** se podrían reescribir para enviar los archivos a cinta.

El comando **tar** puede archivar un subconjunto de archivos mucho más fácil de lo que **dump** puede. La herramienta **dump** sólo trabaja con sistemas de archivos completos a diferencia de **tar**, que puede trabajar con directorios. ¿Acaso ello significa que **tar** es mejor que **dump** para generar copias de seguridad? Bueno, pues en ocasiones sí que lo es...

Ante todo, **dump** resulta ser mucho más eficiente que **tar** cuando requiere respaldar sistemas de archivos completos. Además, **dump** guarda más información sobre los archivos ocupando un poco más de espacio en la cinta; la ventaja es que ello facilita en gran medida el proceso de restauración. Por otro lado, **tar** es verdaderamente multiplataforma: un archivo **tar** creado en Linux lo puede leer el comando **tar** en cualquier otra variante de UNIX. Inclusive, los archivos **tar** procesados por **gzip** ¡los puede leer el popular programa WinZip!

Que usted se sienta mejor con **tar** o con **dump** dependerá de su ambiente y sus necesidades.

NOTA Para más información sobre el uso de **tar**, vea el capítulo 5.

RESUMEN

El tema de las copias de seguridad es uno de los más importantes aspectos sobre mantenimiento de sistemas. Éstos pueden estar instalados, configurados y atendidos de manera brillante, pero sin copias de seguridad, sus sistemas pueden perderse en un abrir y cerrar de ojos. Piense en su plan de respaldos como si fuera la política de seguros de su centro de cómputo.

Este capítulo cubrió aspectos básicos de las unidades de cinta magnética en Linux, además de algunas de las herramientas que, desde la línea de comandos, permiten controlar dichas unidades a fin de generar copias de seguridad de los datos. Con esta información debería serle posible realizar un respaldo completo de su sistema. Afortunadamente **dump**, **restore** y **tar** no son las únicas opciones para crear copias de seguridad en Linux. También existen varios paquetes comerciales y no comerciales. Por varios años fabricantes como Legato y Veritas han provisto a Linux de sofisticados paquetes que ofrecen soluciones impresionantes. Existen otras soluciones más modestas como bru y Lonetar que son buenas para atender un pequeño grupo de servidores administrados por una sola persona. Paquetes de fuente abierta como Amanda y Dirvish también son opciones viables. De éstos destaca Dirvish, pues está dedicado a generar copias de seguridad en disco duro en vez de cinta magnética.

Sin importar cómo decida realizar la tarea de elaboración de copias de seguridad de sus datos, sólo asegúrese de que lo hace.

ÍNDICE

/
/boot, partición, 26
/boot/grub/grub.conf, archivo, 223
/dev, directorio, 174
/etc/exports, archivo de configuración, 480-482
/etc/fstab, archivo, 168-171
/etc/fstab, formato de, entradas de archivo, 169
/etc/group, archivo, 78, 494
/etc/hosts, archivo, 358
/etc/inittab, archivo, 188-190
/etc/modprobe.conf, archivo, 286
/etc/named.conf, archivo, 367, 371-373, 379, 382
/etc/nsswitch.conf, archivo, 391-392, 506-507
/etc/pam.d/other, archivo, 92
/etc/passwd, archivo, 73-77, 494
/etc/passwd, campos del archivo, 73-75
 Contraseña, campo de la, 74
 GECOS, 75

ID del grupo (GID), 75
ID del usuario (UID), 75
Nombre del usuario, 73
/etc/resolv.conf, archivo, 390-391
/etc/services, archivo, 494
/etc/shadow, archivo, 77-78
/etc/shells, 77
/etc/syslog.conf, archivo, 199
 ejemplo, 204
 formato, 202-204
/etc/xinetd.conf, archivo, 193-199
/etc/yp.conf, edición del archivo, 504-505
/home, directorio, 76
/proc, directorio, 228-229
 ajuste de archivos en el, 229
 reportes comunes y ajustes, 232-236
/proc, entradas, 229-232
/proc/kcore, archivo, 229
/root, en el directorio de inicio, 77
/sys/, directorio de dispositivos, 236
/usr/local/src, directorio, 61

A

A (Address), registro (DNS), 375-376
Acceso otorgado a usuarios, 333-335
Acceso, permisos, 85-86
Acceso remoto a archivos, con Samba, 530-531
Acceso, tipos de, 85-86
access_log, archivo, 427
Acentos inversos, 106-107
ACL (access control lists) listas de control de acceso, 72
Active Directory (AD), Windows, 12
Actividad de disco en grado suficiente como para ser oída, 341
Actualizaciones, cómo evitar las que son innecesarias, 214
Adelgazamiento del servidor, 331
Administración de impresoras, CUPS, 567-569
Administración de usuarios con la GUI, 83-85
Administración de volúmenes, 174-184
Administración y manipulación de archivos, 116-124
Administrador de impresión, 558
Administrador de paquetes RPM para la GUI, 57-60
Administradores de inicialización, 33
Administradores de sistema, TCP/IP para, 241-283
Ajustes finos al desempeño NFS, 488
alias, opción (configuración Apache), 426
Almacenamiento de las entradas de un registro, 350-351
Ambiente BASH
 introducción al, 100-105
 comandos múltiples, 106
 control de trabajos en, 101-102
Análisis del registro cronológico, 350
Ancho de banda, monitoreo con MRTG, 351
Anfitriones
 acuerdos para nomenclatura de, 359
 y redes, 263-265
Apache, instalación de, pruebas, 421
Apache, módulos, 418-419
Apretón de manos de tres vías, 258
Árbol DNS de dos capas de profundidad, 361
Archivo de programas (el binario), 86

Archivo de registro de errores (error_log), Apache, 427-428
Archivo de zona raíz, 363
Archivos
 colección en cinta magnética, 597
 ubicación de, en directorios, 123
Archivos de configuración, 12
Archivos de directorios, 111
Archivos de dispositivo, 174
Archivos de registro cronológico (servicios de correo), 451
Archivos de vaciado, lectura y escritura, 274-275
Archivos normales, 110
Archivos para unidades de cinta magnética, creación de, 590-591
ARP (Address Resolution Protocol), Protocolo de resolución de direcciones, 261
 cómo funciona, 261
 con otros protocolos, 262-263
Atajos, línea de comandos, 105-107
Ataques basados en seguridad local, 328, 329-335
Ataques, manejo, 351-352
Atributos de grupo, modificación con groupmod, 96
Autenticación. *Vea también PAM (Pluggable Authentication Modules)*
 de un servidor Windows, 534-535
 uso de OpenLDAP para, 552-554
authconfig, 507

B

Banderas de paquetes IP, 252
BASH (Bourne Again Shell), 77
Bibliotecas
 configuración en un sistema Linux, 67
 construcción desde código fuente, 66-67
Binario (archivo de programa), 86
bind, paquete, 365-370
bind-utils, paquete, 366
Bomba de bifurcaciones (hará caer a su servidor), 334

Boot Loader Configuration, pantalla, 33-34
Bzip2, 118

▼ C

Cadenas (en Netfilter), 306-309
 administración en Netfilter, 315-316
 política predeterminada para, 316
Capas, TCP/IP, 242-248
Cargadores de inicialización, 138-148
carpald.sh, creación del script, 152
cat, programa, 122, 230
CD (CD-ROM), instalaciones utilizando, 19-20
CD de inicialización/ rescate, creación, 140-141
CD de rescate, creación, 140-141
CD-R para copias de seguridad, 587
chgrp (change group), comando, 113
chgrp, comando, 113
chkconfig, herramienta, 155, 383, 420, 436, 478, 496
chmod (change mode), comando, 86, 113-114, 330, 435
chmod, comando, 86, 113-114, 330, 435
chown (change ownership), comando, 112-113
chown, comando, 112-113
chroot(), llamada al sistema, 336-338
chroot, ambiente, 336
chroot, entorno ejemplo, 337-338
CIDR (classless interdomain routing) enrutado interdominio sin clase, 266
CIFS (Common Internet File System) Sistema común de archivos para Internet, 518
Clases de impresión, 559
Clases de usuarios, 86
Clasificación de mensajes en registros, syslogd, 201-202
Cliente NIS, configuración, pruebas, 510
Cliente y servidor, en TCP, 258
Clientes NIS, configuración, 504-510
CNAME (canonical name) registros (DNS), 377
Código fuente
 la idea de regalar el, 6
 roto, 67
Cola de impresión, 569
Colección de archivos, creación en cinta, 597

Comandos, para encontrarlos, 123
Compresión de archivos (gzip), 118
Concatenación de archivos (cat), 122
Conectividad IMAP, pruebas con SSL, 450
Conexión a un recurso compartido, explicación, 10
Conexiones
 TCP, 259
 uso de DNAT en, 305
 uso de SNAT, 304
Confianza nula, después de un ataque, 352
Configuración de red, 285-299
Configuración del firewall Linux, 35-36, 301-325
Configuración del servidor DNS, 370-377
Configuraciones de hardware, 16
Conflictos de hardware, depuración de, 235
Construcción desde código fuente
 bibliotecas y su, 66-67
 localización de problemas, 66-67
Consultas con RPM, 49-52
Contenido de archivos, mostrar, 230
Contenido HTTP, servido desde directorios de usuarios, 425
Contraseña del superusuario, establecer, 37
Contrasenñas
 cambiar con smbpasswd, 533
 cambiar después de un ataque, 352
 encriptación, 519
 establecer para el usuario raíz, 37
 permitir el uso de contraseñas NULL, 533
 selección de, 74
Control de trabajos en el entorno BASH, 101-102
Convenciones acerca del nombramiento de los discos y particiones, 174
Convenciones de nomenclatura de particiones, 174
Copias de seguridad basadas en cinta magnética, 587-588
Copias de seguridad basadas en discos, 587
Copias de seguridad o respaldos, 585-597
 contabilización de los datos que se van a respaldar, 586-587
 desempeño de la red y, 588
dump, herramienta para, 592-595

gestión de cintas magnéticas, 589-592
 herramientas para la línea de comandos, 592-597
 Correo electrónico mediante POP, 348
 Correo electrónico, enviar y recibir con SMTP y POP, 443
 cp, comando, 116
 cp (copy file), comando, 116
 Criptografía de llave pública, 454-457
 cron, programa, 204-206
 crontab, archivo, 205-206
 Cuenta de usuario sin privilegios, 42
CUPS (Common Unix Printing System) Sistema de impresión común de UNIX, 558
 aceptación y rechazo de trabajos de impresión, 567-568
 activación y desactivación de impresoras, 567
 administración de privilegios de impresión, 568
 administración ordinaria, 567-569
 agregar impresoras, 561-566
 clases de impresión, 559
 configuración de la impresora predeterminada, 567
 dispositivos URI que pueden ser configurados, 562-563
 eliminación de impresoras, 568
 impresoras locales e impresoras remotas, 562-563
 instalación, 559-560
 operación, 559
 cupsd.conf, directrices en el archivo, 560-561

D

DAP (Directory Access Protocol) Protocolo de acceso a directorios, 538
 DataFellows (F-Secure) SSH, 457
 db_load, programa, 409
 db4-utils, paquete, 409
 Demonios, procesos, explicación, 192
 Demonios Samba, 519-520
 Dependencias, manejo RPM de, 53-54

Desempeño de la red, copias de seguridad o respaldos y el, 588
 Desinstalación de software, uso de RPM para, 55
 Desmontaje de sistemas de archivo, 168
 Detener tráfico de la red después de un ataque, 352
 dev, directorio (/ dev), 235
 df (disk free), comando, 124
 df, comando, 124, 340
 dhclient, opciones desde la línea de comandos de, 584
 dhclient, comando, 583-584
DHCP (Dynamic Host Configuration Protocol) Protocolo de configuración dinámica de anfitriones, 573-584
DHCP, software
 instalación desde código fuente, 576
 instalación mediante RPM, 575
 DHCP, archivo de configuración, 581-582
 DHCP, cliente, configuración, 583-584
 DHCP, periodo de préstamo en, 574
 DHCP, servidor, 574, 575-582
 comportamiento general durante la operación, 582
 configuración, 575-582
 declaraciones, 577-578
 group, declaración, 577
 host, declaración, 577
 opciones, 581
 parámetros, 578-581
 range, declaración, 578
 shared-network, declaración, 577
 subnet, declaración, 578
 DHCP, solicitudes en, 574
 dhcpd.conf, archivo, 581-582
 dhcpd.conf, opciones comunes del archivo, 582
 dig (domain information gopher) rastreador de información sobre dominios (DNS), herramienta, 384-386
 Dirección de retorno, explicación, 264
 Direcciones IP
 asignación hecha por Linux, 298-299
 octetos de las, 264
 utilizando ifconfig para, 287-292
 Directorio de dispositivos (/sys/devices), 236

Directorio de trabajo, actual, 119
Directorio donde se ubica un archivo, encontrar el, 123
Directorio inicial
 cambio de ubicación, 125-131
 explicación, 75-77
Directorio raíz (/), 24
Directorios, 111
 creación, 119
 eliminación, 119
Discos
 añadir en Linux, 173
 montaje cruzado, 485, 488
 sincronización, 124
 y volúmenes físicos y volúmenes lógicos, 176
Discos locales, montaje y desmontaje, 165
Disk Druid, herramienta de partición, 25
Disk Setup, pantalla (durante la instalación), 27, 32
Disponibilidad de servicios de correo, 450-451
Dispositivos de bloqueo, 111-112
Dispositivos de caracteres, 112
Distorsión IP (asignación de alias), 288-289
Distribuciones de Linux, 4
Distribuciones no comerciales de Linux, 4
DIT (directory information tree) árbol de información del directorio, 538
DN (Distinguished name) Nombre diferenciado, 539
DNAT (Destination NAT) NAT destino, 303
DNAT, uso en una conexión, 305
DNS (Domain Name Service) Servicio de nombres de dominio, 357-393
 dominio de búsqueda predeterminado, 390
 dominio raíz, 360
 dominios y anfitriones, 359
 /etc/nsswitch.conf, archivo, 391-392
 /etc/resolv.conf, archivo, 390-391
 funcionamiento, 359-365
 in-addr.arpa, dominio, 363
 nombres de dominio de primer nivel, 360
 nombres de dominio de segundo nivel, 360

nombres de dominio de tercer nivel, 360-361
[NOTFOUND=action], uso de, 392
subdominios, 361-362
tipos de servidores, 363-364
TLD (top-level domains) nombres de dominio de primer nivel, 360
DNS, configuración del cliente, 389-392
DNS, resolver del lado del cliente, 390-392
Documentación del código fuente, lectura de la, 63
DocumentRoot, opción (configuración de Apache), 423
domainname, comando, 497-498
Dominio predeterminado de búsqueda, 390
Dominios
 acuerdos sobre nomenclatura de, 359
 NIS, 496
 Windows, 12
DoS (Denial of Service), ataque de negación del servicio, 252
du, comando, 122-123
du (disk utilization), comando, 122-123
dump, herramienta, 592-594
 omitir el cálculo de la longitud de la cinta, 594
 parámetros, 593
 respaldar un sistema completo, 594-595
DVD (DVD-ROM), instalación mediante, 19-20
DVD-R, para copias de seguridad o respaldos, 587

 E

echo, servicio, habilitar/inhabilitar, 197-199
Editores (herramientas de edición), 134-135
EFF (Electronic Frontier Foundation) Fundación frontera electrónica, 456
Eliminación de grupos y usuarios, 96
Eliminación de un directorio, 119
Emacs, editor, 134-135
Encabezado de paquetes ARP, 262-263
Encabezado de respuesta HTTP, 415
Encabezado de solicitud HTTP, 414

Encabezado IP, 251
 direcciones IP fuente y destino, 253
 Protocol, campo, 253
 Encabezados Ethernet, 249-250
 Encabezados HTTP, 414-415
 Encabezados (paquete TCP/IP), 244, 249-257
 Encontrar un archivo, 117-118
 Encontrar un comando, 123
 Encriptación, 456-457
 Encriptación de bajo grado, 456
 Encriptación de muchos bits, 457
 Enlace a una interfaz, explicación, 347
 Enlaces suaves, 117
 Enrutado (routing) estático, 266, 295-298
 explicación, 267
 limitaciones de, 268
 Enrutado con rutas estáticas, 295-298
 Enrutado dinámico (con RIP), 268-274
 Entradas de documentación (DNS), 377
 Entradas de registros cronológicos, almacenamiento, 350-351
 ErrorLog, opción (configuración de Apache), 424
 Esquema, LDAP, 541
 Estado de los trabajos de impresión, 569
 Estándares, 135-136
 Ethereal, herramienta, 274, 354
 Ethernet, 249-251
 example.org.db, archivo, 381
 example.org.rev, archivo, 381
 expansión del nombre de archivo, 105-107
 exportfs, parámetros del comando, 482
 exports, archivo de configuración (/etc(exports), 480-482
 ext2, sistema de archivos, 165
 ext3, sistema de archivos, 164
 Extensiones, tipos de, 175

▼ F

fdisk, herramienta, 175, 178-179
 fdisk, menú de ayuda, 179
 Fedora Core Linux
 administrador de paquetes GUI, 58

apagado de servicios en, 349
 herramienta de administración de usuario
 GUI, 84
 herramienta de configuración del servicio
 GUI, 156
 instalación de un servidor DNS, 365-370
 pantalla de registro, 43
 Fedora Core Project, 7
 FHS (File Hierarchy Standard), 136
 find, comando, 117-118
 fips, herramienta, 18
 Firewall, configuración, 301-325
 Firewall Configuration, pantalla (durante la instalación), 35-36
 Firewall, tecnología, 302
 Flujos (conexiones entre cliente y servidor), 304
 FORWARD, cadena (en Netfilter), 308
 FQDN (Fully qualified domain name) Nombre de dominio completamente calificado, 359
 Fragmentos IP, 252
 FreeSSH para Windows, 459
 fsck (File System Check), comprobación del sistema de archivos, herramienta, 158-159, 164, 171-172
 fsck, herramienta, 158-159, 164, 171-172
 FTP en modo pasivo, 397
 FTP (file transfer protocol) protocolo de transferencia de archivos, 19, 395-412
 interacciones cliente / servidor, 396-398
 puerto de control y puerto de datos, 397
 vsftpd, 398-411
 FTP NAT, 306

▼ G

GCOS, explicación, 75
 GET, comando HTTP, 415
 Getent, herramienta, 407
 Ghostscript, 558
 GID (Group ID) Identificadores de grupo, 72
 Gigabit Ethernet, 243
 GNU (GNU's Not UNIX), software, 4
 búsqueda de documentación, 62-63
 compilación, 64-65

compilación e instalación, 60-66
configuración, 63
instalación, 65
limpieza, 66
obtención y desempacado, 61-62
pruebas, 65-66
GNU Public License (GPL) Licencia Pública
GNU, 4, 6, 210
gnuplot, herramienta, 279-281
GPL (GNU Public License) Licencia Pública
GNU, 4, 6, 210
Graficación de ISN en una conexión TCP, 279-
280
group, archivo (/etc/group), 494
Group, opción (configuración de Apache), 424
groupadd, comando, 82-83, 95
groupadd, opciones del comando, 83
groupdel, comando, 82, 96
groupmod, comando, 83, 96
GRUB (cargador de inicialización), 138-148
 acuerdos utilizados en, 139-140
 configuración, 144-145
 etapa 1, 139
 etapa 2, 139
 instalación, 140
 instalación desde el shell GRUB, 141-142
 instalación en el MBR mediante el uso del
 disco flexible GRUB, 143-144
 nuevo núcleo con el cual se puede arran-
 car, 145-148
GRUB, disco flexible de inicialización, 142-143
grub.conf, archivo (/boot/grub/grub.conf),
 223
Grupos
 creación con groupadd, 95
 eliminación con groupdel, 96
 explicación, 72
GUI y el núcleo, separación de la, 9-10
Guion medio (-), utilizado con el comando tar,
 126
gzip, programa, 118
 archivo comprimido con gzip, 121
 vs. el programa WinZip para Windows,
 61

▼ H

HCL (hardware compatibility list) lista de com-
patibilidad de hardware, 16
hello, comando, 65
Herramientas de documentación, 107-109
Herramientas de impresión, del lado del clien-
te, 569-571
Herramientas para DNS, 383-389
host, herramienta (DNS), 383-384
hosts, archivo (/etc/hosts), 358-359
HTTP (Hypertext Transfer Protocol) Protocolo
para transferencia de hipertexto, 248, 414-417

▼ I

ICMP (Internet Control Message Protocol)
Protocolo de mensajes de control de Internet,
248
Identificadores de archivo viciados (NFS), 489
IDS (intrusion-detection system) sistema de
detección de intrusiones, 353
ifconfig, programa, 287-292
 asignación de alias IP, 288-289
 formato del comando, 291
 opciones del comando, 291
imake, comando, 67
IMAP (Interactive Mail Access Protocol) Proto-
colo de acceso interactivo al correo, 441-452
IMAP, modos de acceso, 442
 modo desconectado, 443
 modo en línea, 443
 modo fuera de línea, 443
Impresión, 557-572. *Vea también CUPS (Com-
mon Unix Printing System)*
 con el sistema Linux, 569-571
 terminología, 558
Impresora predeterminada, configuración en
CUPS, 567
Impresoras
 activación y desactivación, 567
 añadir a CUPS, 561-566
 eliminación, 568

in-addr.arpa, dominio (DNS), 363
include, directorio, archivos en el, 313
include, opción (configuración de Apache), 424
include, statement (BIND), 368-369
inetd, programa, 191-193
inetd, servicios, apagar, 348
Inhabilitación de servicios innecesarios, 470
Inhabilitación de un servicio, 157-158
Inicialización, 148-149
 nivel de ejecución correcto para la, 332-333
 en modo monousuario (recuperación), 159
Inicialización del núcleo, 222-223
Inicialización y apagado, 137-160
init, proceso, 149-150, 188
init, servicio, 188-191
inittab, archivo (/etc/inittab), 188-191
inittab, archivo, opciones de los campos de acción, 190-191
inittab, entradas, 189
I-nodes en el sistema de archivos ext2, 163
I-nodes, 162-163
INPUT, cadena (en Netfilter), 308-309
Instalación de CUPS, 559-560
Instalación de Fedora Core Linux, 4, 20-43
 Add Partition, cuadro de diálogo, 27-29
 advertencia no privilegiada al usuario, 42
 Boot Loader Configuration, pantalla, 33-34
 configuración inicial del sistema, 40-43
 Disk Setup, pantalla, 27, 32
 Display, sección, 41-42
 esquema de particiones, 24
 estructura de las particiones en el disco, 23-33
 Finish Setup, pantalla, 43
 Firewall Configuration, pantalla, 35-36
 inicio, 21
 Installation Is Complete, pantalla, 40
 Installation Type, pantalla, 23
 Installing Packages, pantalla, 39
 introducción de información sobre particiones, 25
 Language Selection, pantalla, 22

Make Logical Volume, cuadro de diálogo, 29-31
Make LVM Volume Group, cuadro de diálogo, 30-32
Network Configuration, pantalla, 34-35
Package Group Selection, pantalla, 38
particiones, 26
prerrequisitos, 20
Time Zone Selection, pantalla, 36-37
Welcome, pantalla, 40-41
Instalación de ISC DHCP desde código fuente, 576
Instalación de Linux como un servidor, 15-44
Instalación de Netfilter, 309-313
Instalación de OpenLDAP, 542
Instalación de OpenSSH desde código fuente, 459-461
Instalación de OpenSSH mediante RPM, 462
Instalación de Samba, 520-523
Instalación de Samba desde código fuente, 521-522
Instalación de software, 45-68
Instalación de software utilizando RPM, 52-55
Instalación de un servidor DNS, 365-370
Instalación de un servidor UW-IMAP y un servidor POP3, 446-448
Instalación del núcleo, 220-222
Instalación del servidor de correo Postfix, 433-436
Instalación del servidor DNS, 365-370
 características específicas, 367
 comentarios, 368
 include, statement, 368-369
 logging, statement, 369
 named.conf, archivo, 367
 palabras clave del statement, 368
 programa BIND, 365-366
 qué se instala, 366-367
 server, statement, 369
 zone, statement, 370
Instalación del servidor HTTP Apache, 417-419
Instalación del software DHCP desde código fuente, 576
Instalación del software DHCP mediante RPM, 575

Instalaciones basadas en red, protocolos para, 19
Installation Type, pantalla, 23
Installing Packages, pantalla, 39
Interacciones cliente/servidor (FTP), 396-398
Interfaces de módulos y redes, 286-287
Interfaces de red
 enlace con, 347
 módulos e, 286-287
Interfaz, enlace hacia una, 347
Internet Engineering Task Force (IETF) Fuerza de tareas de ingeniería en Internet, 454
Interrupción de servicios, 348-349
intr, opción (NFS), 488
IP (Internet Protocol) Protocolo de Internet, 246-247, 251-253
 ruta predeterminada, 262
 tcpdump y el, 253
IP privada, explicación, 264
IPP (Internet Printing Protocol) Protocolo de impresión por Internet, 562-563
IPTables, compilación, 312-313
iptables, herramienta, 302, 312-313, 315-322
ISC BIND, instalación del software, 366
ISC DHCP, configuración del servidor, 575-582
ISC DHCP, demonio cliente (dhclient), 583-584
ISNs (Initial Sequence Numbers) Números Iniciales de Secuencia, graficación, 279-280
ISO capa 2, interruptores, 253

▼ J

Joe, programa (editor), 135
Jumbo Ethernet, marcos, 243, 291

▼ K

kill, comando, 130-131
 asuntos de seguridad, 131
 ejemplos de uso, 131
 listado de señales disponibles, 132
known_hosts, archivo (SSH), 469

▼ L

Language Selection, pantalla (durante la instalación), 22
Launch Terminal, 49
LBS (Linux Standard Base Specification), 136
LDAP (Lightweight Directory Access Protocol)
 Protocolo ligero de acceso a directorios, 537-555. *Vea también OpenLDAP*
 creación de entradas de directorios, 549-550
 implementaciones populares de, 538
 modelo cliente/servidor, 539-540
 terminología, 540-541
 usos de, 540
 LDAP, árbol, 539
 LDAP, atributos, 540
 LDAP, directorio, 538, 550-552
 LDAP, entrada, 540
 LDAP, esquema, 541
 LDAP, módulos, 544
 LDAP, objectClass, 541
 LDIF (LDAP Data Interchange Format) Formato de intercambio de datos LDAP, 541, 549
LILO (Linux Loader), 148
Línea de comandos, 99-136
Linux NFS. *Vea NFS (Network File System)*
Linux, diseño de servidor, 16-18
Linux, instalación de. *Vea Instalación de Linux Fedora Core 4*
Linux, núcleo de. *Vea Núcleo*
Linux, sistema Netfilter. *Vea Netfilter*
Linux vs. Windows, 8-13
Lista de procesos, interactiva, 129-130
Lista interactiva de procesos (top), 129-130
Listado de archivos (ls), comando, 109-110
Listado de procesos (ps), comando, 127-129
Listado de procesos utilizando top, 129-130
Listas de envío, 340-341
Listen, opción (configuración de Apache), 422
Llamadas al sistema, 228
Llave de sesión, 456
Llave pública/llave privada, combinación de, 454, 456
Llaves (encriptación), explicación, 456-457
ln (vinculación de archivos), comando, 117

LoadModule, opción (configuración de Apache), 423
 Localhost (sistema local), 390
 localhost.db, archivo, 380
 logging, statement (BIND), 369
 Login, pantalla (Fedora Linux), 43
 LogLevel, opción (configuración de Apache), 426
 Lost+found, directorio, 173
 LPD (Line Printer Daemon) Demonio de impresora en línea, 563
 lpq, comando, 570-571
 lpr, comando, 570
 lprm, comando, 571
 ls, comando, 109-110
 ls, opciones comunes del comando, 110
 LV (Logical Volume) Volumen Lógico, 175
 creación, 175-177, 182-184
 nomenclatura, 183
 visualización, 182-183
 y discos y volúmenes físicos, 176
 LVM, herramientas, 177
 LVM (Logical Volume Management) Administración de volúmenes lógicos, 174, 177

▼ M

MAC (Media Access Control) Control de acceso a medios, direcciones, 249
 Mail delivery agent (MDA) Agente de entrega de correo, 432, 442
 Mail transfer agent (MTA) Agente de transferencia de correo, 432, 442
 Mail user agent (MUA) Agente del usuario de correo, 432, 442
 main.cf archivo del servidor de correo electrónico Postfix, 437-438
 mail_spool, directorio, 437
 mydestination, parámetro, 437
 mydomain, parámetro, 436
 myhostname, parámetro, 436
 mynetworks, variable, 437
 myorigin, parámetro, 436-437
 smtpd_banner, variable, 438
 make, herramienta, 64, 67, 498-499, 502

Makefile, 500-501
 Makefile, archivo, edición, 498-502
 Makefile, errores, 503-504
 man, comando, 107-109
 MAN (metro area networks) redes de área metropolitana, 246
 man, secciones de página, 108
 Manejadores de señal, 130
 Mangle, tabla (Netfilter), 302
 Mapas NIS, 494, 504
 Marcos
 bajo Ethernet, 243
 explicación, 242
 Marcos Ethernet, 243, 291
 Masquerading (SNAT), 303
 MaxClients, opción (configuración de Apache), 423
 Maximum Segment Size (MSS) Tamaño máximo de segmento, 321
 MBR, copia de seguridad o respaldo del, 140
 MDA (mail delivery agent) agente de entrega de correo, 432, 442
 Medios de almacenamiento para copias de seguridad, cómo seleccionar, 587-588
 Métodos abreviados en la línea de comandos, 105-107
 Métodos de instalación (instalación de Linux), 19-20
 Micronúcleo, 9
 MindTerm (cliente SSH), 459
 mkdir, comando, 119
 mkfs.ext3, herramienta, 184
 mknod, comando, 590
 Modo activo de FTP, 397
 Modo de recuperación, inicialización en el, 159
 Modo monousuario (recuperación), inicialización en, 159
 modprobe.conf, archivo (/etc/modprobe.conf), 286
 Monitoreo de ancho de banda con MRTG, 351
 Monitoreo del sistema, 339-341, 350-351
 Monousuario, filosofía, 8
 Montaje
 cruzado de discos, 485, 488
 explicación, 10
 y acceso a una partición, 476

Montaje de particiones, 11
Montaje de recursos remotos compartidos en Samba, 532
Montaje duro vs. montaje suave, 485
Montaje suave vs. montaje duro, 485
Montaje y desmontaje de discos locales, 165-171
Montaje y desmontaje de sistemas de archivos, 474
Montajes, duros vs. suaves, 485
more, comando, 122
mount, comando, 166-168, 485
mount, opciones del comando, 167, 486-487
mount, opciones para NFS, 486-487
Mover directorio inicial y directorios de usuarios, 125-131
MRTG, monitoreo de ancho de banda con, 351-352
mt, parámetros del programa, 591
mt, programa, uso para control de unidades de cinta magnética, 591-592
Multitareas, 9
mv, comando, 116-117
mv (move files), comando, 116-117
MX (Mail Exchanger), registro (DNS), 376

N

named.conf, archivo (/etc/named.conf), 367, 379, 382
 zona de caché en el, 373
 zona primaria en el, 371
 zona secundaria en el, 372
named, proceso, 382
NAT (Network Address Translation) Traducción de direcciones de red, 302
 base, 303-306
 de tres líneas de Rusty, 322-323
 ejemplos de, 303-305
 protocolos amigables para, 306
 seguimiento de conexiones y, 305-306
Necesidades, evaluación de, copias de seguridad, 586-589
Nessus, sistema, 353-354

Netfilter
 activación en el núcleo, 310-312
 ajustes configurados por la instalación, 313
 almacenamiento de ajustes de configuración, 314-315
 cadenas predefinidas en, 306-309
 configuración, 313-322
 definición de rule-spec, 317-319
 extensiones rule-spec con pareja, 319-322
 funcionamiento, 302-309
 icmp, módulo, 319
 instalación si no está presente, 309-313
 iptables, uso de, 315-322
 limit, módulo, 320
 manejo de cadenas, 315-316
 soluciones, 322-325
 state, módulo, 320
 tablas, 302
 tcp, módulo, 320-321
 tcpmss, módulo, 321
 udp, módulo, 322
Netmasks, 265-266
Netstat -an, comando, 340
Netstat, programa, 295, 345-348
Netstat, salida del programa, seguridad de la, 346-347
Network Configuration, pantalla (durante la instalación), 34-35
Network Neighborhood, 10
new-kernel-pkg, comando, 222
NFS, características de versión, 475
NFS (Network File System) Sistema de archivos en red, 10-11, 473-492
 activación, 477
 componentes de, 478-479
 configuración de cliente, 484-488, 490-491
 configuración del servidor, 480-484, 490-491
 consideraciones de seguridad para, 476
 intr, opción, 488
 localización de problemas con permisos denegados, 489
 localización de problemas con identificadores de archivos viciados, 489

- localización de problemas del lado del cliente, 489
localización de problemas del lado del servidor, 484
montaje y desmontaje de sistemas de archivos, 474
opciones de montaje para, 486-487
optimización de desempeño, 488
para almacenar directorios iniciales, 492
para almacenar programas populares, 491
para colas compartidas de correo electrónico, 492
puntos clave, 492
rsize (read size), opción, 488
soporte del núcleo para, 479-480
usos comunes para, 491-492
versiones de, 475
wslice (write size), opción, 488
nfsd, programa, 479
NFSv2, características, 475
NFSv3, características, 475
NFSv4, características, 475
NICs (network interface cards) tarjetas de interfaz de red, 245
 puesta en marcha durante la inicialización, 289
 puesta en marcha en Fedora Core 3, 289-290
 puesta en marcha en Red Hat Enterprise, 289-290
Niñitos del script, 328
NIS, actualización de mapas, 504
NIS, dominios, 496
NIS, herramientas, 512-513
NIS (Network Information Service) Servicio de información en red, 493-515
 all, entrada, 501
 antes y después, 509
 combinación de contraseñas ocultas con contraseñas reales, 500
 combinación de contraseñas ocultas de grupo con grupos reales, 500
 demonios y procesos, 495
 designación de nombres de archivos, 500
 ejemplos de uso del, 508-509
 en redes más grandes que edificios, 515
 en un segmento de red, 514-515
 en una red pequeña, 514
 herramientas GUI para, 507
 inicio, 498
 número mínimo de UID y GID, 499-500
 para archivos compartidos populares, 495
 puntos clave a recordar, 515
 uso en archivos de configuración, 513
Nivel de ejecución, 149
Nivel de ejecución con el cual se puede inicializar, 332-333
Nivel de ejecución, edición, 157
Nivel de ejecución, lista, revisando un programa, 156
Niveles de vaciado, 592
nmap, programa, 353
nmbd, demonio, 520
Nombre de dominio de tercer nivel, 360-361
Nombre de dominio, establecer para NIS, 497-498
Nombre de dominio NIS
 configuración, 510
 establecer, 497-498
Nombres de dominio de primer nivel, 360
Nombres de usuario y contraseñas, 518-519
NS (Name Server), registro (DNS), 375
NSA (National Security Agency) Agencia de Seguridad Nacional, 339, 457
nslookup, herramienta (DNS), 386-387
nss_ldap, módulo, 544
nsswitch.conf, archivo (/etc/nsswitch.conf), 391-392, 506-507
nsupdate, herramienta (DNS), 388
NTFS (Windows), 19
NTLM (Windows), 12
Núcleo
 activación de Netfilter en, 310-312
 adición, 145-148
 cómo se acomoda en un sistema complejo, 211
 como un programa no trivial, 4
 compilación, 219-226
 configuración, 216-219
 construcción, 213-224
 diseño monolítico vs. diseño micro, 9
 evitar actualizaciones innecesarias al, 214

explicado, 4, 210-211
inicialización, 222-223
instalación, 220-222
obtener la versión correcta del, 212-213
parchado, 224-226
preparación para la configuración, 215
que no inicializa, 223
Núcleo, carga del, 149
Núcleo, código fuente
desempacar, 213
encontrar, 211-213
Núcleo, configuración del, 216-219
Núcleo, diferencias del, 5
Núcleo, ejecución del, 149
Núcleo, módulos, 219
Núcleo monolítico, 9
Núcleo, soporte para NFS, 479-480
Núcleo y GUI, separación del, 9-10
NULL, contraseñas, permitir, 533
Número de identificación de paquete IP, 252
Números de puerto, importancia de los, 344

▼ 0

Octeto de una dirección IP, 264
Opciones de configuración, 63
Open Shortest Path First (OSPF) Abrir primero la trayectoria más corta, 273
OpenBSD, el proyecto, 458
OpenLDAP, 541-554. *Vea también* LDAP (Lightweight Directory Access Protocol)
base de datos de segundo plano, 546
configuración, 543-548
configuración de cliente, 548-554
configuración de servidor, 552-553
demonios del lado del servidor, 541
herramientas, 542-543
instalación, 542
slapd, demonio, 544-548
uso para autenticación de usuario, 552-554

OpenSSH
descarga de la última versión de, 460
instalación desde código fuente, 459-461
instalación mediante RPM, 462

que se envía con Linux, 459
uso de, 464
OpenSSH, el proyecto, 458
OpenSSH para MacOS X, 459
OpenSSH, trucos desde el shell, 467
OSI, modelo de referencia, 245-248
 capa 1 (capa física), 246
 capa 2 (capa de enlace de datos), 246
 capa 4 (capa de transporte), 247
 capa 8, 248
 capas 5 a 7, 248
 IP y capa 3 (capa de red), 246-247
 TCP y UDP, 247
OUTPUT, cadena (en Netfilter), 309

▼ P

Package Group Selection, pantalla (durante la instalación), 38
Página "Jobs" de la página Web para administración de CUPS, 569
Página "Printers" en la página Web para administración de CUPS, 566
Página Web para administración de CUPS, 562
 añadir impresora mediante la, 563-566
 administración de impresoras mediante la, 568-569
 página "Jobs", 569
 página "Printers", 566
pam_ldap, módulo, 544
PAM (Pluggable Authentication Modules) Módulos enchufables de autenticación, 13, 86-93
 archivos y su ubicación, 87
 argumentos de configuración, 91
 banderas de control, 90
 configuración, 88-92
 configuración de acceso, 92-93
 depuración, 93
 directorios importantes en, 88
 ejemplo de un archivo de configuración, 90-92
 funcionamiento, 87
 para nombres de usuario y contraseñas, 519
 tipos de módulos, 89

- Paquetes
a través de cadenas NAT, 306
en una red Ethernet, 243
explicación, 242
ruta de paso en una red Linux, 244
tamaño promedio de los, 280-282
- Paquetes, encabezados de (TCP/IP), 244, 249-257
- Paquetes, uso de RPM para validación de, 57
- Paquetes, uso de RPM para verificación de, 56-57
- Parchar al núcleo, 224-226
- Parches, 224-226
descarga e instalación, 224-226
retroceder después de la aplicación de, 226
- parted, herramienta, 175
- Partición de discos usando Fedora, 23-33
- Partición primaria (raíz), 24
- Particiones
creación, 178-180
montaje y acceso, 476
montaje, 11
panorama general de las, 173-174
- Particiones y volúmenes lógicos, creación, 175-184
- PartitionMagic, 19
- passwd, archivo (/etc/passwd), 73-77, 494
- passwd, campos del archivo, 73-75
- PDL (page description language) lenguaje para descripción de páginas, 558
- Perl, para nombres de usuario y contraseñas, 519
- Permisos, 85-86, 115
combinaciones comunes de, 114
NFS, 481
- Permisos de archivos, 115
- pico, programa, 135
- Pilas, TCP/IP, 242
- POP e IMAP, 441-452
- POP (Post Office Protocol) Protocolo de oficina de correos, 348, 441-452
el concepto, 442
verificación, 444
- Portmap, servicio, 477, 490
- postfix, script, 435
- Postfix, servidor de correo electrónico
configuración, 436-438
instalación, 433-436
instalación desde código fuente, 434-436
instalación mediante RPM, 433
liberar la cola de correo, 439
localización de problemas, 440
newaliases, comando, 439
operación, 438-440
registro de correo, 440
revisar la cola de correo, 439
- POSTROUTING, cadena (en Netfilter), 309
- PostScript, 558
- PREROUTING, cadena (en Netfilter), 308
- Primaria, zona en el archivo names.conf (DNS), 371
- Primarios, servidores (DNS), 363
- Privilegios de impresión, administración, 568
- Proc, sistema de archivos, 227-237
ajustes en archivos de, 229
/proc, directorio, 228-229
/proc, entradas, 229-232
/proc/kcore, archivo, 229
reportes comunes y parámetros ajustables, 232-236
- Procesos
iniciar y detener los, innecesarios, 331-332
listado, 127-129
señalización, 130-132
- Procesos innecesarios, detener la operación de, 331-332
- Programa no trivial, núcleo como, 4
- Programas que se ejecutan como raíz, 333
- Protocolo HTTP, 19
- Protocolo NFS, 19
- Protocolos de instalación basados en red, 19
- Protocolos para instalaciones basadas en red, 19
- Pruebas de una instalación de Apache, 421
- ps, comando, 127-129, 330
opciones comunes, 127
ps -af, comando, 188
ps auxww, 340
salida muestra, 128-129
- ps, salida del comando, encabezados de columna, 129

Pseudoterminal, lanzamiento, 49
PTR (pointer), registro (DNS), 376
Puerta de acceso, explicación, 292
Puerto de control (FTP), 397
Puerto de datos (FTP), 397
Puertos
 HTTP, 415-416
 TCP y UDP, 344
Puertos destino, 344
Puertos fuente, 344
Pull (extracción desde servidores), 496
Punto de choque de la red, 303
Puntos de repartición, en Windows, 10
Push (propagación hacia servidores), 495
PuTTY para Win32, 458
PV (Physical Volume) Volumen físico, 174
 asignación de un volumen a un grupo, 181-182
 creación, 180-181
pvcreate, comando, 180
pvdisplay, comando, 180
pwd, comando, 119

Q

Qt, entorno de desarrollo, 217
Quién está registrado en el sistema
 who, comando, 133
 whois, comando, 387-388

R

RAID, sistemas para copias de seguridad o respaldos, 587
Raíz, contraseña del usuario, establecer, 37
Raíz, dominio (DNS), 360
Raíz, programas que se ejecutan como, 333
Raíz, usuario, explicación, 72
RARP (Reverse ARP) ARP a la inversa, 263
rc, escritura de scripts, 151-152
rc, scripts, 150-155
Reconocimiento puro (en TCP), 260
Recuperación, velocidad de, 588-589
Recurso compartido, enlace a un, 10

Recursos limitados, 334-335
Red Hat Enterprise Linux (RHEL) series, 7
Red Hat Linux
 apagado de servicios en, 349
 liberación comercial de, 7
 puesta en marcha de NIC en, 289-290
Red Hat Package Manager. *Vea RPM (Red Hat Package Manager)* Administrador de Paquetes Red Hat
Redes, anfitriones y, 263-265
Redes conectadas por un router, 267
Redes IP, juntando, 263-273
Redirección, 104-105
Redirección de puertos con SSH, 465-466
Referencia hacia adelante (DNS), 371
Referencia indirecta, 162
Referencias sobre criptografía, 457
Registro cronológico, 339-340
Registro de Windows vs. archivos de texto en Linux, 11-12
Registro diario, explicación, 164
Registros DNS, tipos de, 373-377
ReiserFS, sistema de archivos, 164
Resolución DNS invertida, 363-364
Resolución hacia adelante (DNS), 363
resolv.conf, archivo (/etc/resolv.conf), 390-391
Resolver (cliente DNS), 390-392
restore, herramienta, 595-596
 opciones desde la línea de comandos, 596
 restauración completa, 595-596
 restauración típica, 595
Riesgo, mitigar el, 336-339
RIP
 enrutado dinámico con, 268-274
 red de un campo más pequeño que utiliza, 269
RIP, algoritmo de, 273
rm, comando, 66
rmdir, comando, 119
rndc, programa (DNS), 388-389
route, comando, 294-295, 297
Router de Linux, simple, 295-298
Router
 dos redes conectadas por un, 267
 explicación, 292
 simple, 295-298

RP, registros (DNS), 377
RPC, procesos que facilitan el servicio NFS, 478-479
RPC, programas, 478
RPC, servicios, 477
rpc.gssd, proceso, 479
rpc.idmapd, proceso, 479
rpc.lockd, proceso, 479
rpc.mountd, proceso, 479
rpc.nfsd, proceso, 479
rpc.rquotad, proceso, 478
rpc.statd, proceso, 478
rpc.svcgssd, proceso, 479
rpcinfo, comando, 477, 498
RPCs (remote procedure calls) llamadas a procedimientos remotos, 474
RPM, archivo, 46
RPM, base de datos, 379
rpm, comando, 379, 462
RPM (Red Hat Package Manager) Administrador de paquetes Red Hat, 46-60

- búsqueda de documentación de software, 62-63
- compilación de software, 64-65
- configuración de software, 63
- consulta con, 49-52
- consulta de software desde otras fuentes, 52
- consulta de todos los paquetes, 50
- consulta sobre los detalles de un paquete en específico, 50-51
- instalación de software DHCP mediante, 575
- interfaz GUI para, 57-60
- limpieza de software, 66
- obtención y desempacado de software, 61-62
- opciones comunes, 48
- para instalación y desinstalación de software, 46
- pruebas de software, 65-66
- uso para desinstalación de software, 55
- uso para instalación de OpenSSH, 462
- uso para instalación del servidor de correo de Postfix, 433
- uso para instalar software, 52-55

uso para validación de paquetes, 57
uso para verificar paquetes, 56-57
RPM, transacción, 54
RPM, verificación de atributos de error, 57
rule-spec, extensiones con pareja (Netfilter), 319-322
rule-spec (Netfilter), 317-322
Russell, Rusty, 322
Rusty, NAT de tres líneas de, 322-323
Ruta predeterminada

- en IP, 262
- explicación, 292

Rutas, 292-298

 **S**

Salida de un programa como parámetro de otro programa, 106-107
Samba, 517-536

- administración, 523
- inicio y apagado, 523
- instalación, 520-523
- instalación y compilación desde código fuente, 521-522
- localización de problemas, 535-536
- montaje de recursos compartidos remotos, 532
- opciones de configuración, 522
- para autenticación en un servidor Windows, 534-535
- para nombres de usuario y contraseñas, 519

scp (Secure Copy), programa, 464, 468
ScriptAlias, opción (configuración de Apache), 426
scripts de configuración, 64
Scripts para el inicio

- creación, 153-155
- explicación, 76

scsidev, comando, 590
SecureCRT para Windows, 459
Seguimiento del rastro de conexión

- con estado, 305
- NAT y el, 305-306

- Segundo nivel, nombres de dominio de, 360
Seguridad
 como la mitigación del riesgo, 329
 de la salida de netstat, 346-347
 de servidores de correo electrónico, 432
 del comando kill, 131
 en la red, 343-354
 local, 327-342
 para NFS, 476
 titularidad de procesos y, 416-417
 y el eslabón más débil, 459
Seguridad, herramientas, 352-354
Seguridad de la red, 343-354
 herramientas para la, 352-354
 TCP/IP y la, 344-345
Seguridad SSL, 449-450
SELinux (Security-Enhanced Linux) Linux con seguridad mejorada, 339
sendmail, implicaciones de seguridad de, 432
Señal, enviar a un proceso, 130-132
server, statement (BIND), 369
ServerAdmin, opción (configuración de Apache), 423
ServerName, opción (configuración de Apache), 422-423
ServerRoot, opción (configuración de Apache), 422
service, comando, 405
Services, archivo (/etc/services), 494
Servicio NFS
 inicio y apagado del, 477-478
 procesos RPC que facilitan el, 478-479
Servicios
 activación y desactivación, 155-158, 197-199
 apagado, 348-349
 con números de puerto bien conocidos, 345
 configuración, 156
 desactivación, 157-158, 470
 seguimiento, 345-349
Servicios bien conocidos, explicación, 345
Servicios centrales del sistema, 187-207
Servicios de correo electrónico
 archivos de registro cronológico, 451
disponibilidad de, 450-451
temas de, 449-451
Servicios innecesarios, desactivación de, 470
Servidor, 9
 energía de respaldo para un, 17
 instalación de Linux como un, 15-44
 optimización del desempeño de un, 17
 protección y seguridad de un, 17
 tiempo activo, 18
Servidor Apache HTTP
 archivos de configuración, 422
 cómo procesa la propiedad, 416-417
 configuración, 421-427
 inicio durante el arranque, 420
 inicio y apagado, 419-420
 instalación, 417-419
 instalación desde código fuente, 418
 localización de problemas, 427
 opciones de configuración comunes, 422-424, 426-427
 página simple a nivel raíz, 421-422
 puesta en marcha, 413-428
Servidor BIND (Berkeley Internet Name Domain) Dominio de Nombres de Internet de Berkeley, 365
 archivo de configuración, 367-370
 archivos de la base de datos, 378-383
 instalación, 366
 valores de tiempo abreviados, 381
Servidor con autoridad (DNS), 363
Servidor de correo electrónico
 configuración, 436-438
 ejecución, 439-440
 implicaciones de seguridad, 432
 instalación, 433-436
 instalación desde código fuente, 434-436
 instalación mediante RPM, 433
 localización de problemas, 440
 registros cronológicos de correo, 440
 vaciado de la cola de correo electrónico, 439
 verificación de la cola de correo electrónico, 439
Servidor, diseño de, 16-18

Servidor DNS
 investigación de lentitud, 277-278
 puesta en marcha, 379-383
Servidor FTP para accesos anónimos, puesta en marcha, 407
Servidor FTP
 configurado como sólo anónimo, 407-408
 configurado con usuarios virtuales, 408-409
 inicio y pruebas, 400, 404-406
 personalización, 407-411
Servidor, herramienta de configuración, 508
Servidor IMAP
 instalación, 446-448
 uso de Telnet para conectarse a, 445
Servidor NFS, herramienta de configuración del, 483
Servidor NIS maestro, 495-496
Servidor NIS maestro, poner en marcha la propagación hacia servidores esclavos, 511
Servidor NIS, configuración, 496-504
Servidor POP3
 instalación, 446-448
 uso de Telnet para conectarse al, 444-445
Servidor SSH, inicio y apagado, 462-463
Servidor UW-IMAP, instalación, 446-448
Servidor Web. *Vea Servidor HTTP Apache*
Servidores caché, explicación, 364
Servidores de alto volumen, consideraciones, 234
Servidores de nombre raíz (DNS), 363
Servidores DNS secundarios, 363
Servidores NIS, 495-496
Servidores NIS secundarios, 495, 510-512
Servidores NIS secundarios (esclavos), designación, 499
Sesión, en TCP, 247
SetGID, programa, 86
SetUID, programa, 86, 329-331
sftp, herramienta para la línea de comandos, 468-469
Shells, explicación, 77
showmount, comando, 483-484
SID (Security Identifier) Identificador de seguridad, 339

SIDs (system IDs) Identificadores de sistema, 72
Sincronización de discos (sync), 124
Sistema de archivos
 administración, 165-173
 creación, 184-186
 desmontaje, 168
 /etc/fstab, archivo, 168-171
 montaje y desmontaje de discos locales, 165-171
 selección, 165
Sistema de inicialización dual, 18-19
Sistemas de archivos, 161-186. *Vea también NFS*
 (Network File System) Sistema de Archivos en Red
 estructura de los, 162-165
 montaje y desmontaje, 474
 slapd, demonio (OpenLDAP), 544-548
 SMB/CIFS, conjunto de protocolos, 518
 SMB, protocolo, 19
 SMB (Server Message Block) Bloque de mensajes del servidor, 19, 518, 563
 smbclient, herramienta, 518, 529-531
 acceso a archivos remotos, 530-531
 navegación de un servidor, 530
 smbd, demonio, 519
 smbpasswd, comando, 533
 SMTP (Simple Mail Transfer Protocol) Protocolo de transferencia de correo simple, 429-440
 SNAT (Source NAT) NAT Fuente, 303
 SNAT, uso en una conexión, 304
 snoop, herramienta, 274
 Snort, intrusion-detection system (IDS) Sistema de detección de intrusiones, 353
 SOA (Start of Authority), registro (DNS), 374-375
 Software gratuito, 5-8
 Software, instalación de, 45-68
 sort, comando, 281
 src, directorio (/usr/local/src), 61
 ssh, cliente, 464
 ssh, comando, secuencias de escape respaldadas, 467
 ssh, oferentes de clientes, 458-459
 SSH (Secure Shell), 453-470
 archivos utilizados por, 469

características principales de, 456
redirección de puertos con, 465-466
versiones y distribuciones de, 457-459

SSH, túneles, 348, 464-469
sshd, demonio, 462, 465
sshd_config, archivo, 463, 469
SSL, 248

Stallman, Richard, 5-6

statement, palabra clave, 367

su, comando, 133-134

su (switch user), comando, 133-134

Subdominios (DNS), 361-362

Subredes, 264-265

Suma de verificación, encabezado TCP, 253, 256

Superbloques, 163-164

Superusuario, 72

SuSE Linux, GUI

- editor de nivel de ejecución, 157
- para administración de discos, 184
- para administración de paquetes, 58-59
- para administración de usuarios y grupos, 85

SWAT (Samba Web Administration Tool)

- Herramienta Web para administración de Samba, 524-527
 - creación de un recurso compartido, 527-529
 - menús, 526-527
 - puesta en marcha, 524-526

SYN, protección contra la inundación, 233-234

SYNACK, 233

sync, comando, 124

syncookie, 233

SysFS (system file system) sistema de archivos *system*, 235

syslog, 350-351

syslog.conf, archivo (/etc/syslog.conf), 199

- formato del, 202-204
- muestra, 204

syslogd, demonio, 199-204

- clasificaciones de los mensajes de registro cronológico, 201-203
- invocación, 199
- niveles de prioridad de los mensajes de registro cronológico, 201-202
- parámetros, 200

ubicación de entradas de los mensajes de registro cronológico, 203

valor del servicio de mensajes de registro cronológico, 201

System.map, archivo (cap!), 221

system-config-authentication, herramienta, 507

▼ T

Tabla de filtros (Netfilter), 302

Tabla de particiones, mostrar, 180

Tablas de enrutado, 267-268

tail-f, comando, 427-428

tar (tape archive), comando, 61, 119-121, 597

- desde y hacia un dispositivo físico, 121
- opciones, 120-121
- uso de guión medio con el, 126

tarball, 61

TCP

- cliente y servidor, 258
- reconocimiento puro, 260

TCP, apuntador de encabezado urgente, 257

TCP, banderas, 255

TCP, conexión, 258-261

- apertura, 258-259
- cierre, 260-261
- transferencia de datos, 259-260

TCP, encabezado, 254-257

TCP/IP, 241-283

- seguridad de la red y, 344-345
- y el modelo OSI, 245-248

TCP/IP, capas, 242-248

TCP/IP, encabezados de paquete, 244, 249-257

TCP/IP, paquetes

- en una red Ethernet, 243
- explicación, 242
- ruta de paso en una red Linux, 244

TCP/IP, pilas, 242

TCP, paquete, tamaño promedio de un, 280-282

TCP, puertos, 344

TCP (Transmission Control Protocol) Protocolo para control de transmisiones, 247

tcpdump, herramienta, 250, 258-259, 261-262, 274-282, 354

- cómo capturar más por paquete, 275

cómo evitar la captura de paquetes de sesión, 276
impacto en el desempeño, 275
lectura de su página man (manual), 276
lectura y escritura de archivos de vaciado, 274-275
mostrar un trazador de rutas, 276-277
para graficación de ISN en una conexión TCP, 279-280
para investigar un DNS lento, 277-278 y el protocolo IP, 253
telinit, comando, 191
telnet
 a través de Internet, 459
 uso para conectar a un servidor IMAP, 445
 uso para conectar a un servidor POP3, 444-445
Terminal, lanzamiento de una pseudoterminal, 49
textinfo, sistema, 109
Texto plano, 454
Tiempo útil, 18, 211
Time Zone Selection, pantalla (durante la instalación), 36-37
Time-to-live, campo (TTL), paquete IP, 252
Tipos de archivos y directorios, 110
Tipos de directorios, 110-111
Tipos de particiones, códigos hexadecimales para, 179
Titularidad de procesos y seguridad, 416-417
TLDs (Top-level domains) Dominios de primer nivel, 360
top, comando, 129-130
top, salida del comando, 130
Torvalds, Linus, 7
Trabajo de impresión
 aceptación y rechazo de, 567-568
 en pausa, 569
 en progreso, 569
traceroute, programa, 276-277
traceroute, salida, 276-277
Tráfico de la red, alto después de un ataque, 352
Trayectorias múltiples de igual costo, explicación, 273

Tuberías, 104, 112
Tuberías nombradas, 112
Túnel seguro, creación, 464-469
TXT, registros (DNS), 377

▼ U

UDP (User Datagram Protocol) Protocolo para datagramas de usuario, 247
encabezados de paquete, 257
puertos, 344
usos para, 247
UIDs (user IDs) Identificadores de usuario, 72
umount, comando, 168
uname, comando, 132
Unidades de cinta magnética
 administración, 589-592
 para copias de seguridad o respaldos, 587
 mt, programa de control, 591-592
untar, herramienta, 62
uptime, comando, 18
URG, bandera, en el encabezado TCP, 257
URI (Uniform Resource Information) Información de recursos uniformes, 562
User, opción (configuración de Apache), 424
useradd, comando, 79-81, 93-94
useradd, opciones del comando, 80-81
userdel, comando, 82, 96
UserDir, directriz, para ofrecer contenido HTTP, 425
UserDir, opción (configuración de Apache), 424
usermod, comando, 82, 95-96
Usuario normal, explicación, 72
Usuario, atributos, modificación con usermod, 95-96
Usuario, autenticación, utilizando OpenLDAP para, 552-554
Usuarios
 administración, 71-97
 creación con useradd, 93-94
 eliminación con userdel, 96
 explicación, 72
 otorgar permiso de acceso a, 333-335
 permisos de acceso y, 85-86

Usuarios, administración
desde la línea de comandos, 79-83
desde una GUI, 83-85
Usuarios, clases de, 86
Usuarios de red, explicación, 9
Usuarios, herramientas de administración, 79-85
Usuarios, información de, en archivos de texto, 72-73
Usuarios múltiples (multiusuario), explicación, 8
Usuarios, ofrecer contenido HTTP desde directorios de, 425
Usuarios Samba, creación, 532-533

▼ V

Validación de paquetes, uso de RPM para, 57
Variables del ambiente BASH, 102-105
establecer valores en las, 103
impresión, 102-103
quitar valores establecidos, 103-104
Variables del entorno (BASH), 102-105
como parámetros, 105
establecer valores en las, 103
impresión, 102-103
quitar valores establecidos, 103-104
Ventana corrediza, explicación, 256
Verificación de paquetes, uso de RPM para, 56-57
VG (Volume Group), 174-175, 181-182
vgdisplay, comando, 181
vgextend, comando, 181
vi, editor, 134
Vínculos fuertes, 111, 117
Vínculos simbólicos, 111, 154
VirtualHost, opción (configuración de Apache), 426-427
Visualización de archivos, una pantalla a la vez, 122
VPN de los pobres, 464-469
vsftpd (very secure FTP daemon), 398-411
archivos y directorios de configuración, 400
configuración, 399-406

construcción desde el código fuente, 398
obtención e instalación, 398-399
opciones de configuración, 401-404
vsftpd.conf, archivo de configuración, 399-400, 409-410

▼ W

W, comando, 133
WANs (wide area networks) redes de área amplia, 246
wget, herramienta, 62
whereis, comando, 123
which, comando, 123
winbindd, demonio, 520, 534-535
Windows, autenticación en un servidor utilizando Samba, 534-535
Windows, dominios y Active Directory en, 12
Windows, Registro vs. archivos de texto en Linux, 11-12
Windows vs. Linux, 8-13

▼ X

X Window, sistema, 10
xinetd, demonio, 191-199, 448
xinetd, servicios, apagado, 348
xinetd.conf, archivo (/etc/xinetd.conf), 193-199
bloque de instancias, 193
bloque de valores predeterminados, 193
variables, 194-196
XML, 248

▼ Y

yp.conf, archivo (/etc/yp.conf), edición, 504-505
ypbind, activación y desactivación, 505-506
ypinit, herramienta, 497
operación, 511-512
uso, 502-503
ypserv, demonio, 496, 498
Yum (herramienta para administración de paquetes GUI), 59-60

▼ z

Zona de caché en el archivo named.conf (DNS),
373

Zona DNS, 370

Zonas secundarias en el archivo names.conf
(DNS), 372

Una de las metas de este libro es proporcionar una manera para que los administradores de Windows se sientan cómodos con Linux, mediante la demostración del paralelismo de los sistemas. En esta sección se visualizan algunos de los conceptos más importantes para ambos sistemas operativos, de modo que el lector puede ver cómo se establecen sus fundamentos. Además de las comparaciones, hemos visualizado dos procesos comunes en un sistema Linux: el procesamiento de las solicitudes a la Web y el procesamiento de un nuevo correo electrónico.

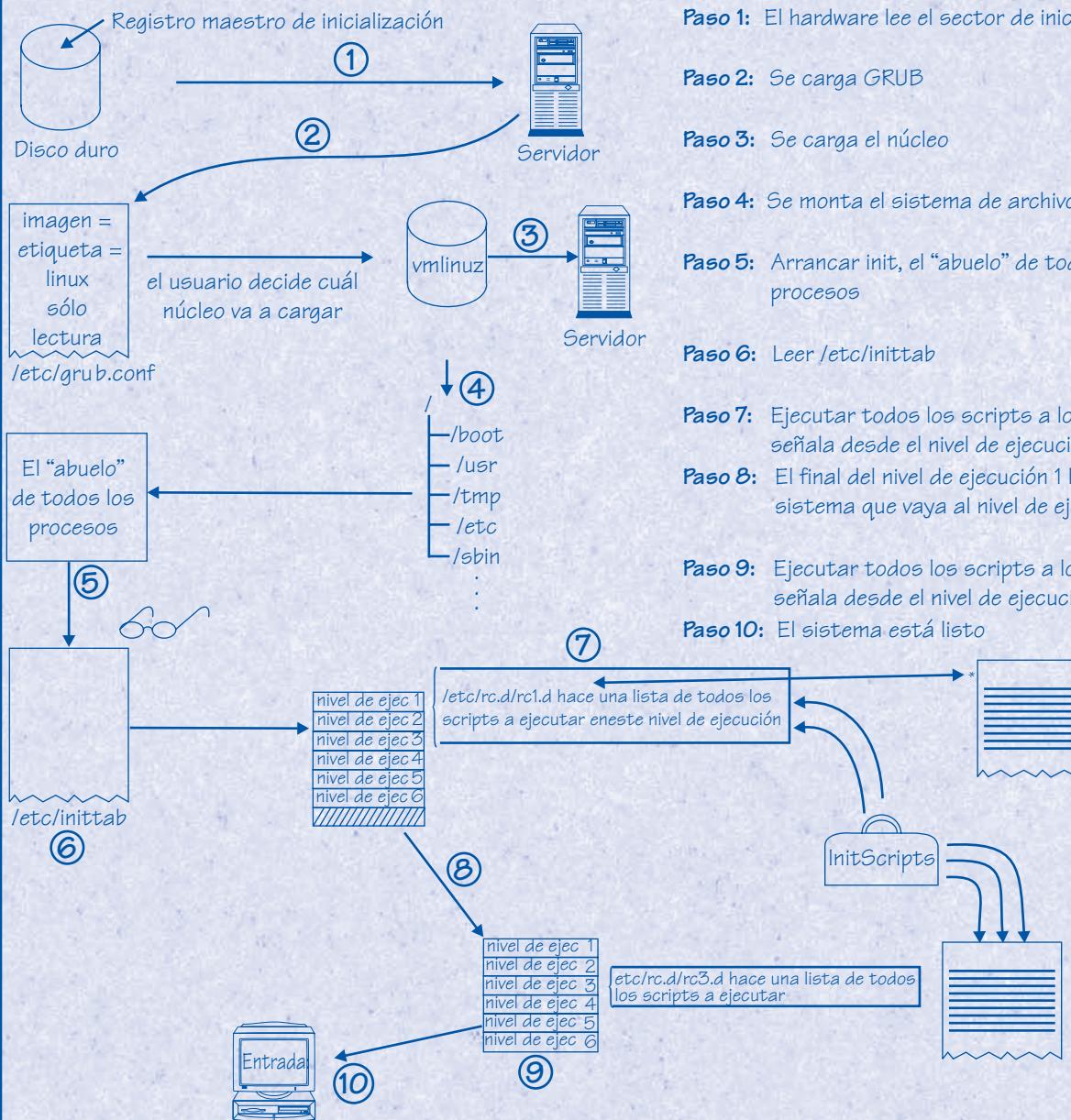
Copias azules de la Administración de Linux

Tabla de contenido

Proceso de inicialización de Linux en comparación con el Servidor Windows 2003.....	2
Red basada en Linux en comparación con el Servidor Windows 2003.....	4
Solicitud a la Web en comparación con la tubería de procesamiento de correo electrónico.....	6
Proceso de paro en Linux en comparación con el Servidor Windows 2003....	8

Proceso de inicialización de Linux en comparación con el Servidor Windows 2003

El proceso de inicialización de Linux



Paso 1: El hardware lee el sector de inicialización

Paso 2: Se carga GRUB

Paso 3: Se carga el núcleo

Paso 4: Se monta el sistema de archivos raíz

Paso 5: Arrancar init, el “abuelo” de todos los procesos

Paso 6: Leer /etc/inittab

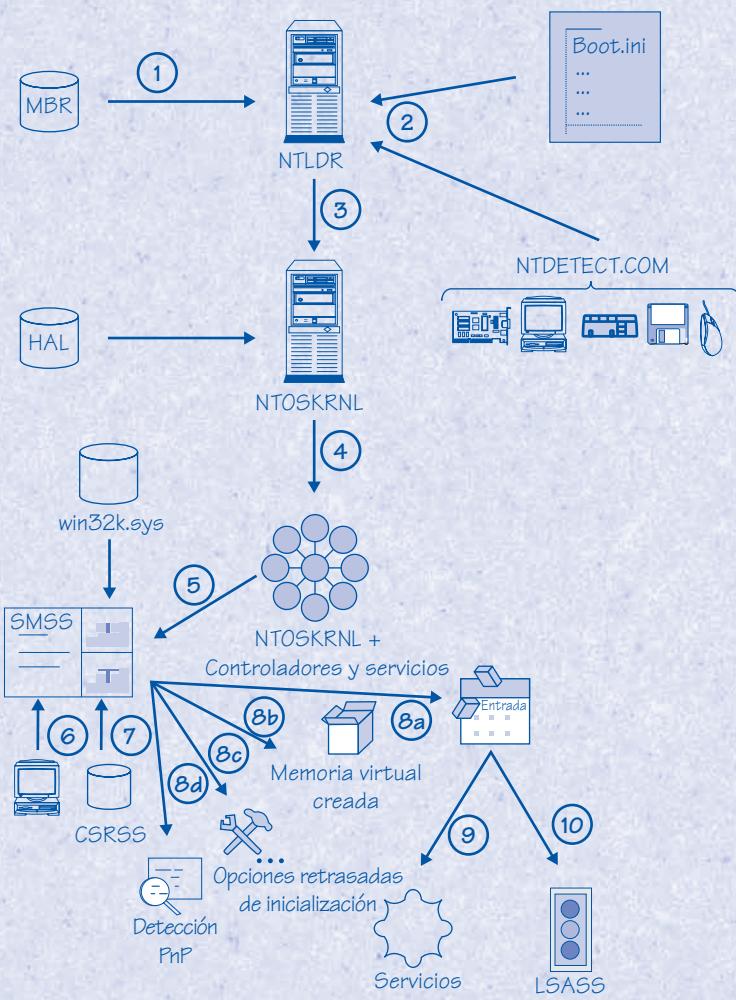
Paso 7: Ejecutar todos los scripts a los que se señala desde el nivel de ejecución 1

Paso 8: El final del nivel de ejecución 1 le dice al sistema que vaya al nivel de ejecución 3

Paso 9: Ejecutar todos los scripts a los que se señala desde el nivel de ejecución 3

Paso 10: El sistema está listo

El proceso de inicialización del Servidor Windows 2003

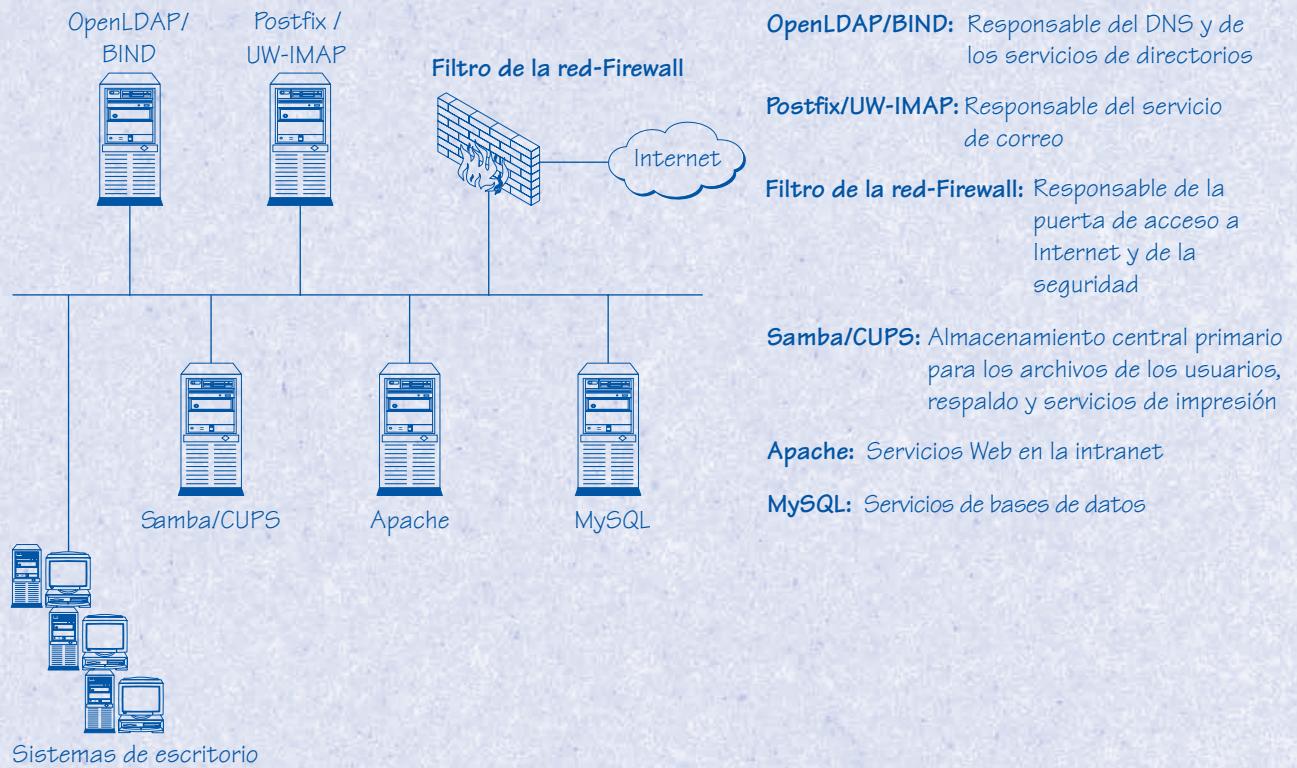


- Paso 1:** Se carga el Registro maestro de inicialización (MBR) y se arranca el NTLDR (Cargador NT)
- Paso 2:** El NTLDR lee BOOT.INI y los resultados de NTDETECT.COM
- Paso 3:** Se carga NTOSKRNL.EXE con la HAL.DLL apropiada
- Paso 4:** Se cargan los controladores y los servicios del núcleo
- Paso 5:** SMSS.EXE (Session Manager, Administrador de sesiones) inicia y crea el entorno, invoca el modo del núcleo (Win32.sys)
- Paso 6:** El modo del núcleo inicia el subsistema gráfico
- Paso 7:** Se inicia CSRSS.EXE (modo del usuario)
- Paso 8a:** Se inicia Winlogon
- Paso 8b:** Se crea la memoria virtual
- Paso 8c:** Se inician otras opciones retrasadas de inicialización (por ejemplo, componentes restantes de un proceso de instalación)
- Paso 8d:** Se realiza la detección PnP
- Paso 9:** Se inicia Services.exe (Service Control Manager, Administrador de control de los servicios)
- Paso 10:** Se inicia lsass.exe (Local Security Authority), Autoridad de seguridad local

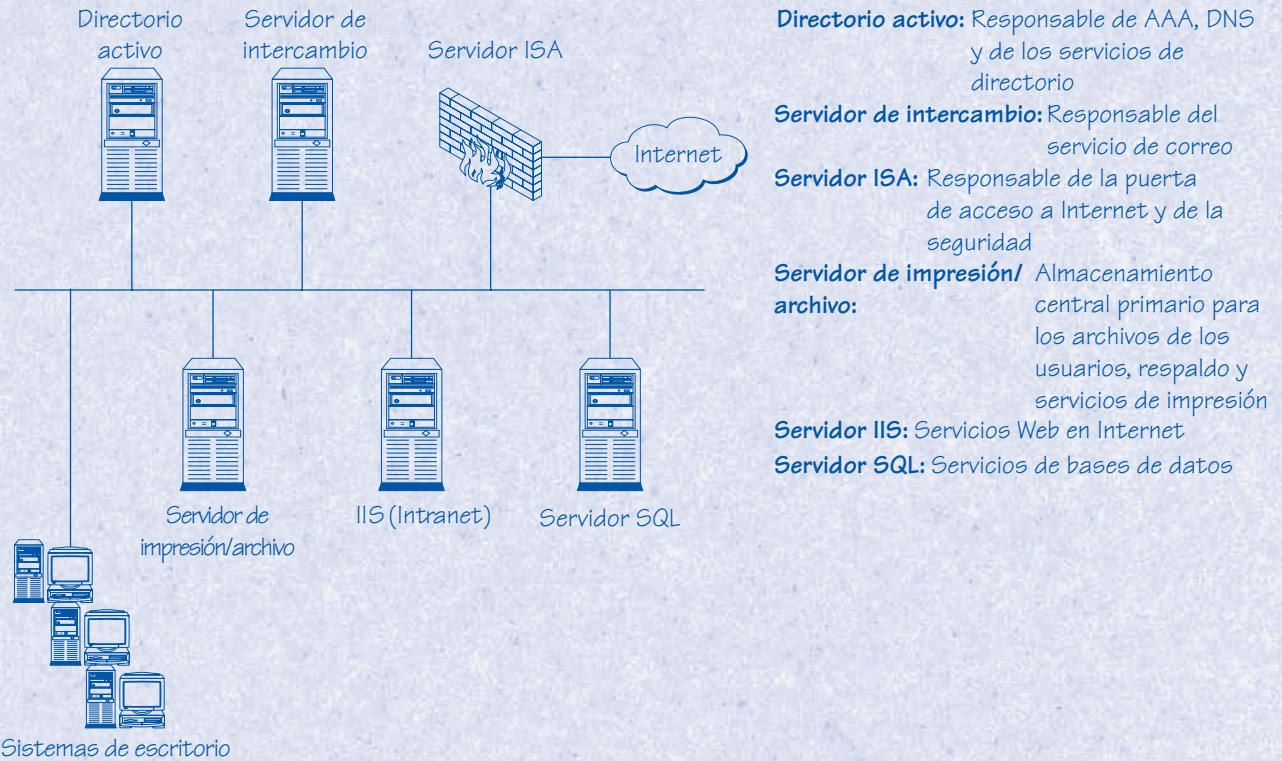
Aun cuando los nombres y el orden exacto de los pasos son diferentes a los del proceso de inicialización de Linux, el principio sigue siendo el mismo en Windows: arrancar un cargador de inicialización, iniciar un núcleo, iniciar los servicios y obtener un mensaje para conceder el acceso. Cuando un usuario entra, inicia programas específicos del usuario. Una diferencia clave es el procedimiento para las inicializaciones controladas, en donde el administrador desea controlar estrechamente lo que se arranca durante las sesiones de detección de fallas. Linux implementa esto con el uso de niveles de ejecución y parámetros del núcleo. Windows implementa este concepto usando “Safe Mode”.

Red basada en Linux en comparación con el Servidor Windows 2003

Red Linux

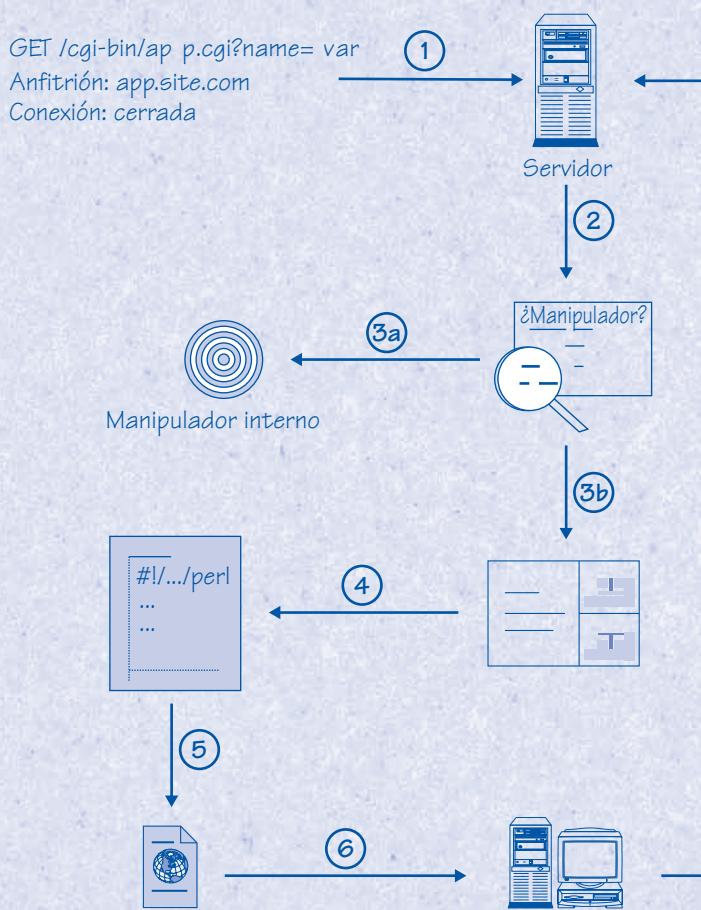


Red basada en el Servidor Windows 2003



Las dos redes son funcionalmente equivalentes: cualquier cosa que pueda hacer Windows, también lo puede hacer Linux. La diferencia clave es la manera en que Linux divide los componentes claves en fragmentos funcionales más pequeños. Desde el punto de vista de un administrador de sistema, la diferencia significa que los servicios de Linux se pueden separar en servidores más pequeños e independientes, para tareas específicas. Tenga presente que es posible consolidar las funciones tanto en Linux como en Windows. Para un entorno de oficina suficientemente pequeño, es posible fusionar muchas de estas funciones en un servidor.

Solicitud a la Web en comparación con la tubería de procesamiento de correo electrónico



Paso 1: Aceptar una nueva conexión HTTP

Paso 2: Procesar la solicitud y determinar el manipulador

Paso 3a: Si el manipulador es interno, procesarla internamente

Paso 3b: Si el manipulador es CGI, crear el entorno para manejar la solicitud

7

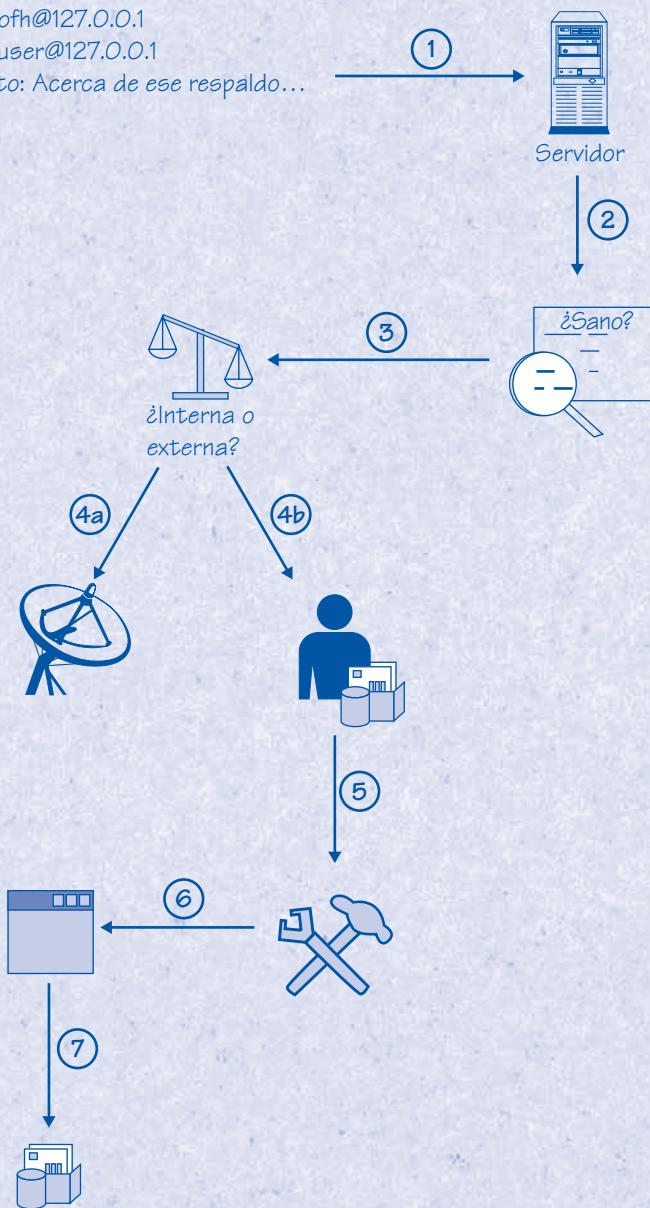
Paso 4: Pasar la solicitud al script CGI

Paso 5: Capturar la salida del script CGI

Paso 6: Entregar la salida del script al cliente

Paso 7: Dependiendo de los ajustes de la solicitud HTTP, cerrar la conexión o esperar por una nueva solicitud

De : bofh@127.0.0.1
Para: user@127.0.0.1
Asunto: Acerca de ese respaldo...

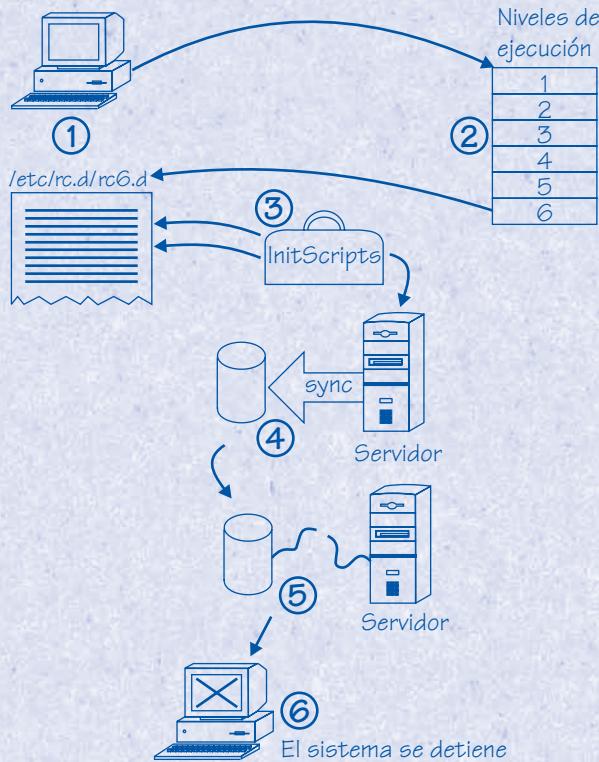


- Paso 1:** El correo electrónico llega al servidor
- Paso 2:** Se comprueba la sensatez inicial en relación con la corrección, información pasada, etcétera
- Paso 3:** ¿Entrega interna o externa?
- Paso 4a:** Si la entrega es externa, dirigir el correo hacia el servidor apropiado de correo
- Paso 4b:** Si la entrega es interna, encontrar el buzón apropiado del usuario
- Paso 5:** Aplicar todos los filtros apropiados (spam, listas de correos, etcétera)
- Paso 6:** Pasar el mensaje al agente de entrega
- Paso 7:** El agente de entrega anexa el correo electrónico al buzón apropiado

El procesamiento del tráfico en la Web y del correo electrónico, dos prácticas muy comunes con Linux, en realidad son procesos bastante complejos. Por debajo se está llevando a cabo una gran cantidad de actividad. Familiarizarse con estos pasos hará que la detección de fallas del proceso sea un poco más fácil la siguiente ocasión en que necesite estudiar un problema difícil.

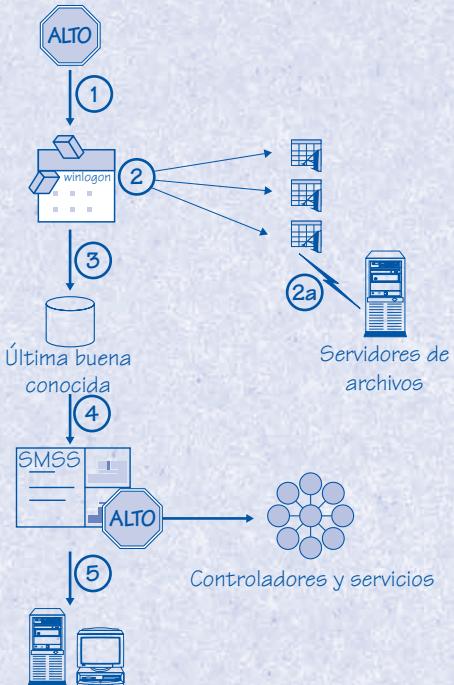
Proceso de paro en Linux en comparación con el Servidor Windows 2003

Paro en Linux



- Paso 1:** Se invoca el comando de paro o reinicio
Paso 2: Se le dice a Init que cambie al nivel 6 de ejecución
Paso 3: Los scripts que están en el nivel 6 llaman a los scripts init con el parámetro “stop” (alto)
Paso 4: Se escriben memorias intermedias pendientes para el disco (`sync'd`)
Paso 5: Se desmontan los sistemas de archivos
Paso 6: El sistema hace alto

Paro en el Servidor Windows 2003



- Paso 1:** Winlogon recibe el mensaje de paro
Paso 2: Winlogon envía a todos los procesos y servicios el mensaje de paro y espera a que se efectúen
Paso 2a: Cualesquier proceso (como Explorer.exe) que tienen conexiones con la red cierran sus conexiones
Paso 3: Se escribe la información “Última buena conocida” en el Registry
Paso 4: SMSS para y suspende todos los servicios del núcleo
Paso 5: El sistema hace alto

Aun cuando puede parecer que los procesos de paro no tienen mucho en común, sus principios sí lo tienen. En donde Linux usa niveles de ejecución para emitir señales de detención hacia todos sus servicios, Windows emite señales de detención a través del tablero. Entonces los dos realizan una limpieza general antes de que, por último, paren los servicios del núcleo.